Advant® OCS
with Master software

Functional Units Part 3
SEQ GROUP

User's Guide

Use of **DANGER**, **WARNING**, **CAUTION**, and **NOTE**

This publication includes, **DANGER**, **WARNING**, **CAUTION**, and **NOTE** information where appropriate to point out safety related or other important information.

DANGER	Hazards which could result in severe personal injury or death
WARNING	Hazards which could result in personal injury
CAUTION	Hazards which could result in equipment or property damage
NOTE	Alerts user to pertinent facts and conditions

Although **DANGER** and **WARNING** hazards are related to personal injury, and **CAUTION** hazards are associated with equipment or property damage, it should be understood that operation of damaged equipment could, under certain operational conditions, result in degraded process performance leading to personal injury or death. Therefore, comply fully with all **DANGER**, **WARNING**, and **CAUTION** notices.

TRADEMARKS

Advant, AdvaCommand, AdvaInform, AdvaBuild, and AdvaControl are registered trademarks of ABB Asea Brown Boveri Ltd. Switzerland

ABB Master, ABB MasterPiece, ABB MasterView, ABB MasterAid and ABB SuperView are registered trademarks of ABB Asea Brown Boveri Ltd. Switzerland.

NOTICE

The information in this document is subject to change without notice and should not be construed as a commitment by ABB Industrial Systems AB. ABB Industrial Systems AB assumes no responsibility for any errors that may appear in this document.

In no event shall ABB Industrial Systems AB be liable for direct, indirect, special, incidental or consequential damages of any nature or kind arising from the use of this document, nor shall ABB Industrial Systems AB be liable for incidental or consequential damages arising from use of any software or hardware described in this document.

This document and parts thereof must not be reproduced or copied without ABB Industrial Systems AB's written permission, and the contents thereof must not be imparted to a third party nor be used for any unauthorized purpose.

The software described in this document is furnished under a license and may be used, copied, or disclosed only in accordance with the terms of such license.

Copyright © ABB Industrial Systems AB 1997

3BSE001264/C

TABLE OF CONTENTS

Chapter 1 - Introduction

1.1 General 1-1
1.2 Structure of the Manual 1-2
1.3 Purpose of the Manual 1-5
1.4 Other documentation 1-5

Chapter 2 - Sequence Control in Advant OCS with Master software

2.1 Standard Representation of Sequences 2-1
2.2 Principles of Realization of a Sequence in Advant OCS with Master software 2-3
2.3 Various Methods of Realizing Sequence Control 2-6

Chapter 3 - SEQ

3.1 Configuration 3-1
3.2 Function 3-3
  3.2.1 Start of Sequence 3-5
  3.2.2 Specification of Step Numbers 3-5
  3.2.3 Normal Execution 3-7
  3.2.4 Time Functions 3-7
  3.2.5 JUMP 3-9
    3.2.5.1 Jump in Sequence 3-9
    3.2.5.2 Conditional Jumps 3-9
  3.2.6 HOLD 3-10
  3.2.7 Manual Execution 3-10
  3.2.8 Clearing 3-10
    3.2.8.1 Clearing of Step 3-10
    3.2.8.2 Clearing of the Sequence 3-12
  3.2.9 POS and END 3-12
  3.2.10 Operator Functions 3-12
3.3 PC Elements 3-15
  3.3.1 PC Element SEQ 3-15
    3.3.1.1 Connections to Operator Station 3-20
  3.3.2 PC element STEP 3-21
3.4 Data Base Element SEQ 3-24
3.5 Operator Functions 3-34
  3.5.1 Presentation 3-34
  3.5.2 Dialog 3-41
  3.5.3 Event Handling 3-46
  3.5.4 Group Alarm 3-49

CONTENTS (continued)

3.6 Application Examples 3-50
  3.6.1 Example 1 3-50
  3.6.2 Example 2 3-51
  3.6.3 Example 3 3-52
  3.6.4 Example 4 3-53
  3.6.5 Example 5 3-56

Chapter 4 - GROUP

4.1 Configuration 4-1
4.2 Function 4-3
4.3 PC Elements 4-4
  4.3.1 PC Element MMC-IND 4-4
  4.3.2 PC Element MMC-ORD 4-8
4.4 Data Base Element MMCX 4-11
4.5 Operator Station Functions 4-26
  4.5.1 Presentation 4-26
  4.5.2 Dialog 4-32
  4.5.3 Event Handling 4-36
  4.5.4 Group Alarm 4-38
4.6 Application Examples 4-39

Chapter 1 Introduction

1.1 General

The function library available in an ABB Advant Controller 400 Series installation includes an extensive range of different data base elements and PC elements. A required control function can be built up with the AMPL language (ABB Master Programming Language).

Functional units are available to supplement the library for more complex functions. The functional unit is a “package” of different program functions (such as PC elements and data base elements).

Each functional unit has a specific communication module for connection to the Operator station (for control and supervision). The operators handling in the Operator Station has a function adapted dialog. Standardized display elements are used for visual indications.

The packaging simplifies documentation and implementation of, for example, functions with both control function and associated operators actions via display screen and keyboard. PC elements and functional units can be used freely together.

Applications include feedback control, sequence control and motor/valve controls.

The functional units are constituted of pre-prepared parts. These consist of different control functions or interfaces to a user designed control function and different operators, alarm and event handling functions.

The descriptions of the functional units for process signal connections, AI, AO, DI, DO include a certain hardware function to give a complete picture of the function. All other functional units are completely implemented in software.

NOTE

General knowledge of the Advant Controller 410/450 and Advant Station 500 Series Operator Station systems is required to fully understand the contents of this manual.

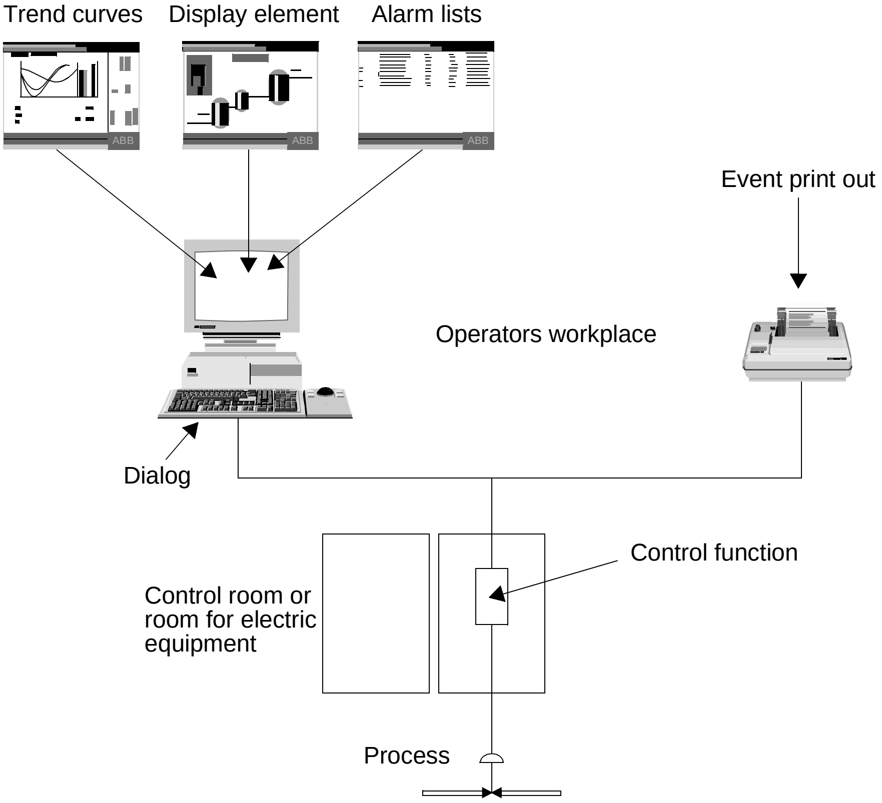


Figure 1-1. Software functions which can be included in a functional unit

1.2 Structure of the Manual

The manual, Functional Units, is for practical reasons divided into seven (7) physical parts. The contents of these seven parts are:

Part 1	Common Properties
Part 2	AI AO DI DO
Part 3	SEQ GROUP
Part 4	PIDCON RATIOSTN MANSTN
Part 5	GENCON GENBIN GENUSD
Part 6	MOTCON VALVECON
Part 7	PIDCONA

Functional Units Part 1 Common Properties, may be read first, in order to define the Functional Unit philosophy. This part also includes descriptions of common properties, valid for all Functional Units.

Part 2 to 7 do not have to be read in part number order. Each part stand by itself (together with part 1). A reader can start with Part 1 and then succeed with the part most urgent.

The contents of the different parts are:

Functional Units Part 1

- General descriptions of the functional units
- Descriptions of common system functions

Functional Units Part 2

- AI (Analog Input). The functional unit AI (Analog Input) incorporates functions for:
  - Scanning of analog and pulse train inputs from the process
  - Filtering, linearization and scaling of such inputs
  - Storage of such inputs or internally calculated analog variables
  - Update blocking and input simulation
  - Limit checking
  - Operator station functions
  - Internal and external alarm detection and handling.
- AO (Analog Output). The functional unit AO (Analog Output) incorporates functions for:
  - Output signal storage and updating
  - Auto and manual modes of control
  - Limit checking prior to storage
  - Scaling and output of an analog control signal to the process
  - Operator station functions.
- DI (Digital Input). The functional unit DI (Digital Input) incorporates functions for:
  - Scanning of digital process inputs.
  - Filtering and inversion of such inputs.
  - Storage of input data
  - Update blocking and input simulation
  - Operator station functions.

- DO (Digital Output). The functional unit DO (Digital Output) incorporates functions for:
  - Output signal storage and updating
  - Auto and Manual modes of control
  - Inversion of stored data prior to output
  - Output of a digital control signal to the process
  - Operator station functions.

Functional Units Part 3

- SEQ (Sequence).
  - SEQ (Sequence) is a software package - a functional unit - designed for universal use in sequence applications.
- GROUP (Group).
  - GROUP (Group) is used for group-start and group-stop of motors, valves and other binary objects coordinated as a group in the process. These objects are normally controlled by MOTCON or VALVECON with which GROUP is compatible.

Functional Units Part 4

- PIDCON
  - PIDCON has been developed for control functions in different processes. It is therefore called a process controller.
- RATIOSTN
  - RATIOSTN is mainly intended for ratio control. This ratio control is performed in cooperation with a process controller PIDCON or PIDCONA.
- MANSTN
  - MANSTN is intended for control of unregulated objects.

Functional Units Part 5

- GENCON
  - GENCON (GENeral object CONtroller) is a general tool which enables the user to create desired control functions. This Functional Unit include a User Defined PC Program.
- GENBIN
  - GENBIN (GENeral object BINary controller) is a general tool which enables the user to create desired control functions. This Functional Unit include a User Defined PC Program.
- GENUSD
  - GENUSD (GENeral object USer Defined controller) is a general tool which enables the user to create desired control functions. This Functional Unit include a User Defined PC Program.

Functional Units Part 6

- MOTCON
  - MOTCON is a functional unit designed for the control of motors in different processes.
- VALVECON
  - VALVECON is a functional unit designed for the control of 2-position valves in different processes.

Functional Units Part 7

- PIDCONA
  - PIDCONA is a complete PID control function with autotuning and adaptive control, including functions for operator interface and event handling.

1.3 Purpose of the Manual

The purpose of this manual is to give the Advant Controller 400 Series implementer information about the powerful library of predefined function units “packages”, in order to provide all the information needed for the use of the different functional units.

Functional Units Part 1, contains the following information:

- General descriptions of the functional units
- Descriptions of common system functions

The other six parts include

- Detailed descriptions of the design and function of the different functional units
- Examples of applications

1.4 Other documentation

In addition to the unified documentation for the functional units offered by these series of Function Units Parts (part 1-7), there are other manuals which document individual types of program functions within a functional unit. There is, for example, a reference manual, *PC Elements Advant Controller 400 Series Reference Manual*, which embraces all of the PC elements included in the controller. The PC elements contained in any particular functional unit are also included in this. A corresponding manual is available for data base elements, *Data Base Elements Advant Controller 400 Series Reference Manual*. Both are valuable aids for practical use in the day-to-day design work, when knowledge of the function is available. Information in this manual is based on the following documents:

- *PC Elements Advant Controller 400 Series Reference Manual*
- *Data Base Elements Advant Controller 400 Series Reference Manual*
- *AdvaCommand Basic Functions User’s Guide*
- *AdvaCommand User Interface Reference Manual*

Chapter 2 Sequence Control in Advant OCS with Master software

Many industrial processes or sub-functions in processes can be described as a sequence of activities. The activities can be completely consecutive in time, or divergent or parallel and can be conditional in different ways.

Control of such step-by-step activities is designated sequential control. Characteristic of sequential control is that it simplifies the solution of problems in a systematic and simple way and permits rapid diagnosis of faults in the process or control system.

2.1 Standard Representation of Sequences

There are different ways of describing sequences. Path/time-diagrams and flow diagrams are traditional methods. Function charts are now used more frequently and have been the subject, for some years, of French standard (GRAFCET) and German standard. IEC has completed development of a standard (IEC 848) based on the above national standards.

Advant OCS with Master software contains PC elements for programming sequences which make it easy to write application programs on the basis of such function charts.

Examples of sequence structures described in accordance with IEC-norms are shown in the figures below. The structures are created in Advant Controller 410/450 with the structure elements SEQ (Sequence header) and STEP (Step element).

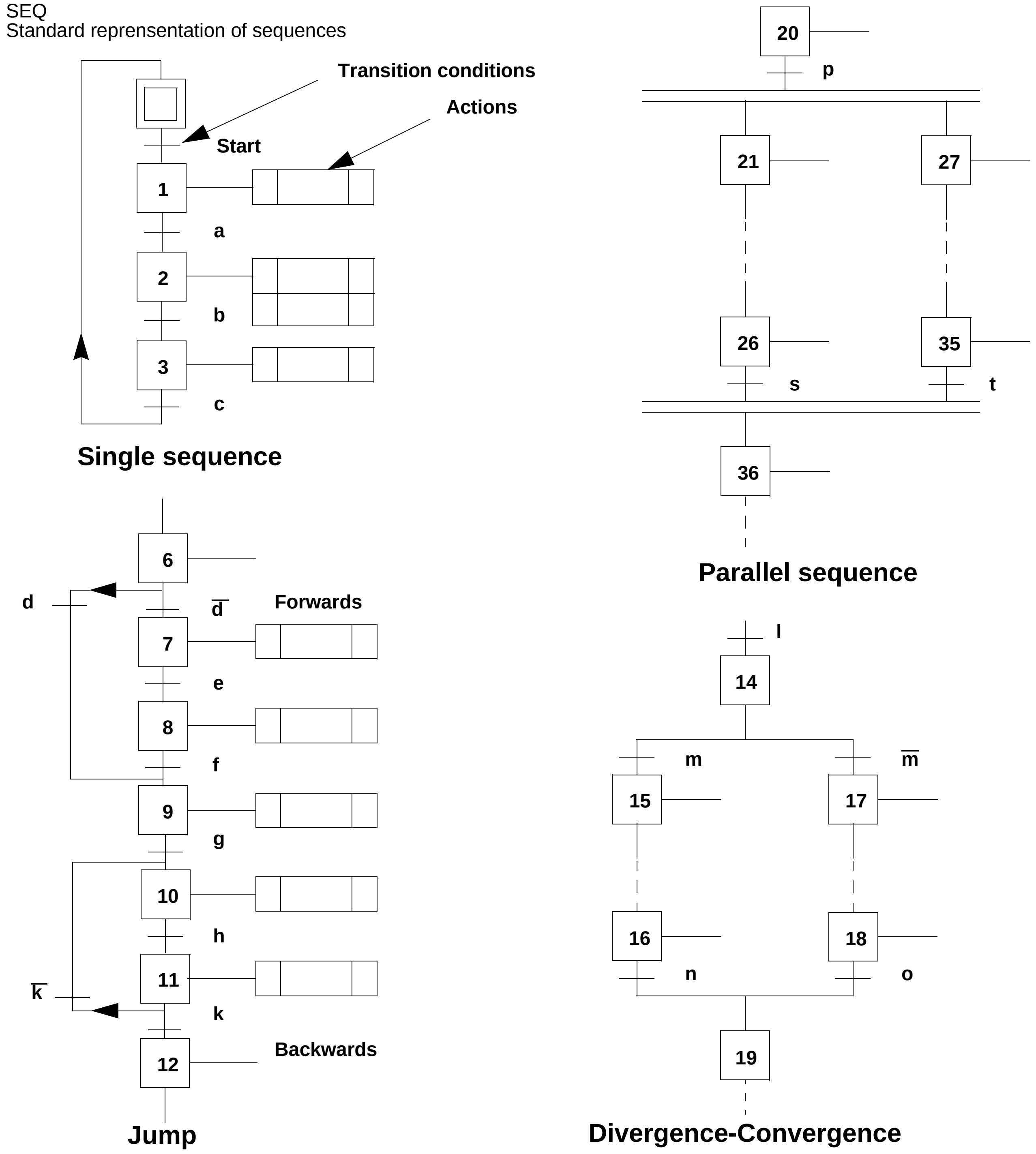


Figure 2-1. Examples of sequence structures

2.2 Principles of Realization of a Sequence in Advant OCS with Master software

How sequence control is realized in Advant OCS with Master software on the basis of function blocks, is illustrated by a simple example.

In the example in Figure 2-2, a wagon carries material from station A to station B. The operator starts movement of the wagon by pressing push-button S1. A winch draws the wagon forward to station B where the material is emptied. After being emptied, the wagon returns to station A for manual refilling.

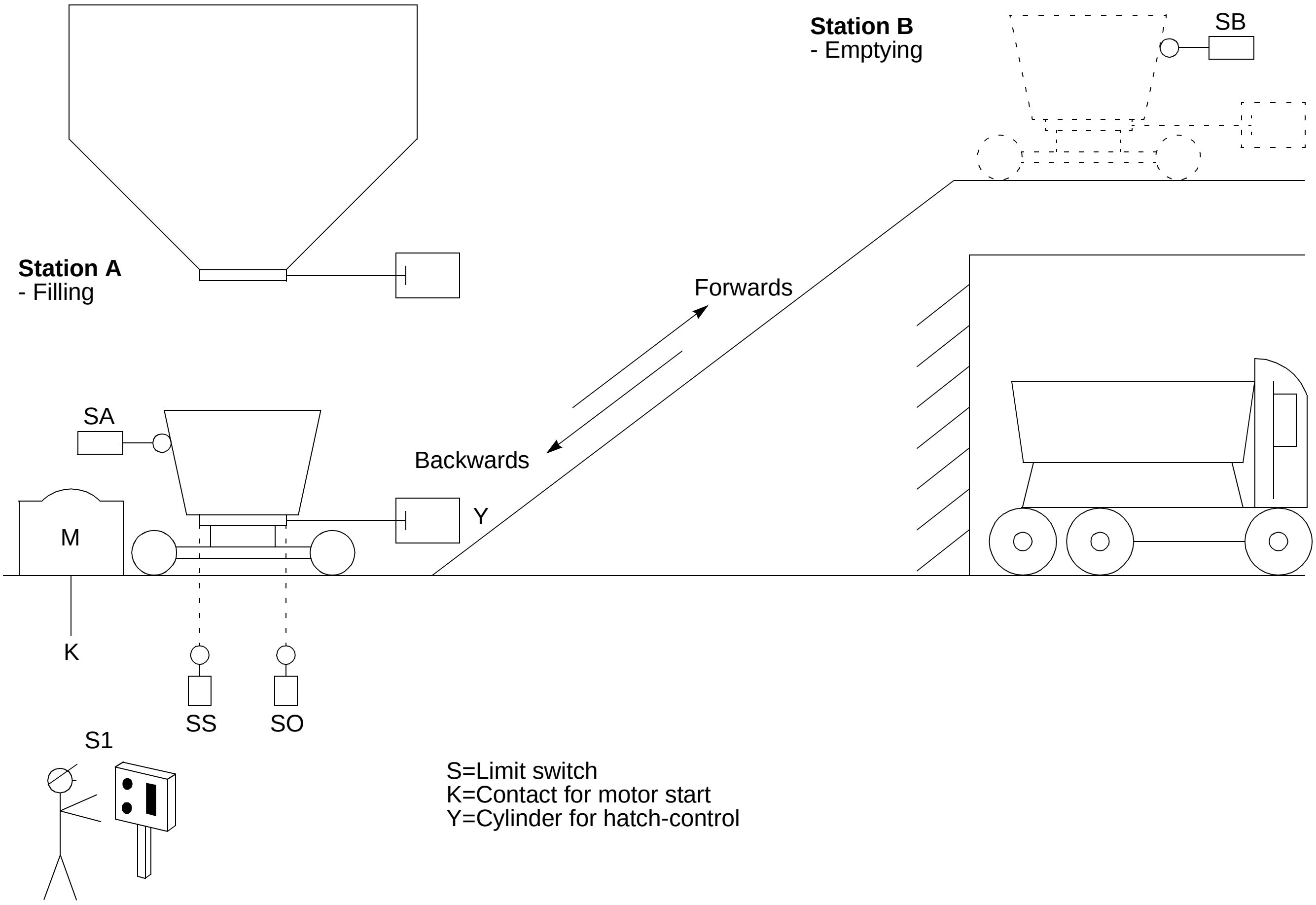


Figure 2-2. Loading of material

The loading process is described with the help of the function charts and the implementation of the control functions in Advant Controller 400 Series is performed with AMPL (ABB Master Programming Language). The relation between these is illustrated by placing the function charts (Figure 2-3) beside the printed function diagram (Figure 2-4).

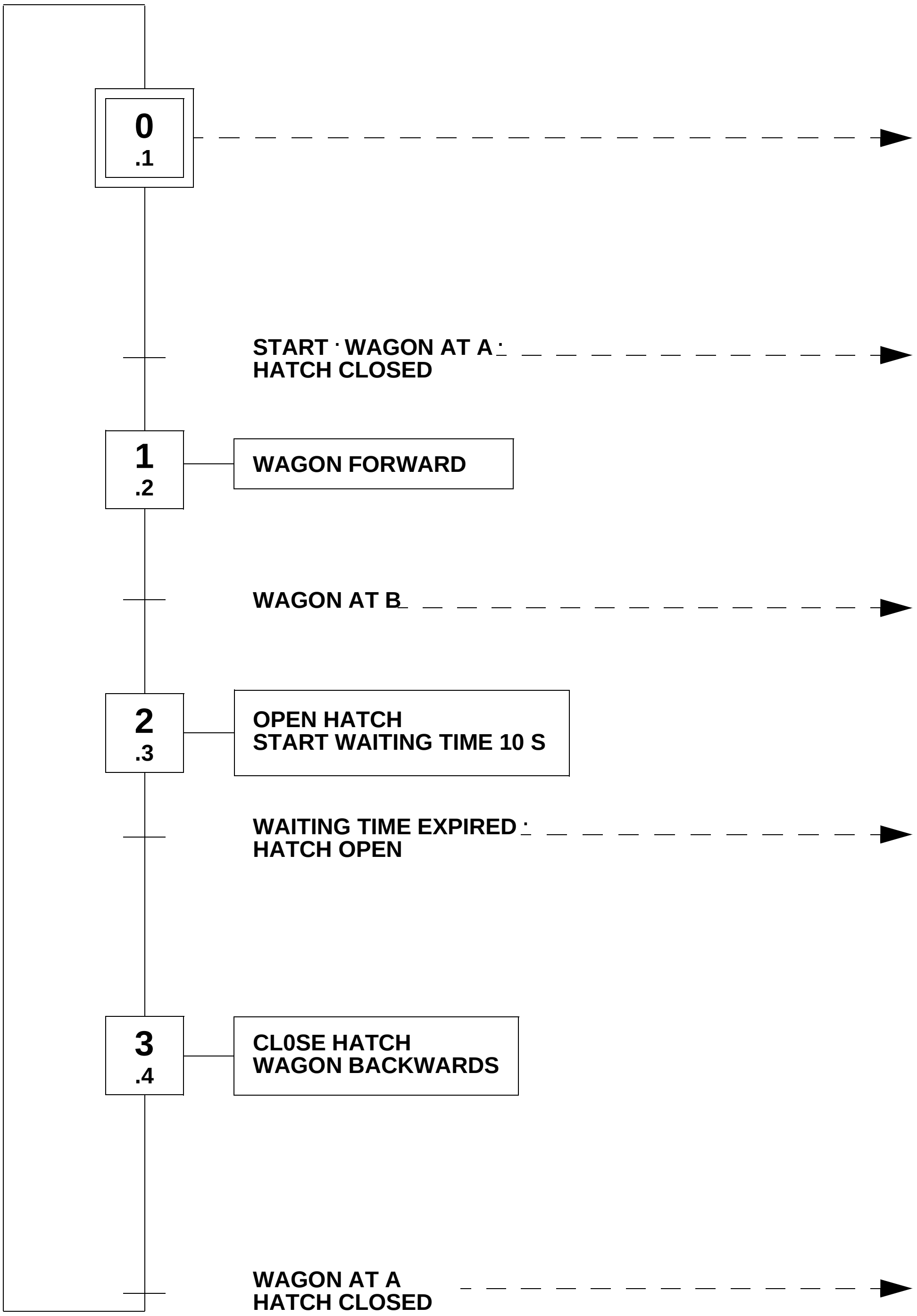


Figure 2-3. Realization of sequence control with AMPL (1 of 2)

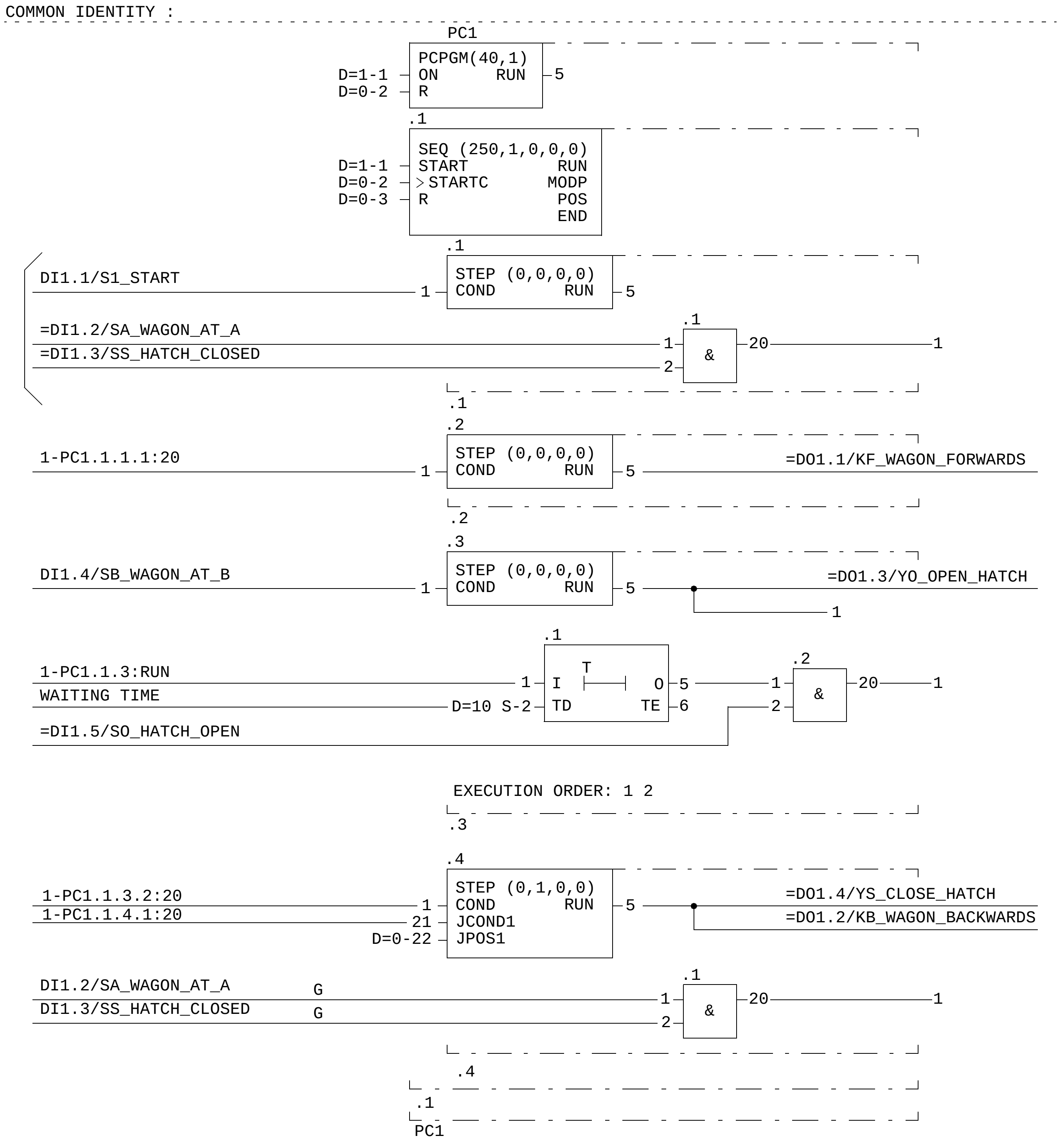


Figure 2-4. Realization of sequence control with AMPL (2 of 2)

2.3 Various Methods of Realizing Sequence Control

In general, all construction of sequence control in Advant OCS with Master software is based on the structure elements SEQ and STEP. Control functions within steps, both numerical functions such as calculations and setpoint generation and logical functions and condition for stage exchange are created with the complete function library as a resource.

There are alternative ways of creating Operator functions. Push buttons and indications in a conventional mimic panel can be connected directly to structure elements and subsidiary program functions via I/O. In applications in which display-screen-based operator stations are used, as the only operator's workplace, or in combination with a conventional local panel, it is advantageous to use one of the functional units SEQ or GROUP. These offer pre-prepared Operator functions with dialog and different display elements for presentation of sequence controls which simplify the application work.

The functional unit SEQ is intended for general use in different types of applications. The dialog and the presentation are closely connected, functionally, to the structure elements SEQ and STEP. The functional unit therefore includes these elements in its description (Chapter 3 SEQ).

This standardized functional unit offers application-oriented operator displays.

Examples of information which are of interest to the operator in connection with sequence control are given below.

- Whether the transition conditions for step are satisfied
- The command signals which are active
- The jump possibilities which are available
- Step descriptions

This information is shown automatically when the programmer has filled in the Operator functions part of the structural element STEP.

Figure 2-5 illustrates how the presentation is structured. The process diagram shows broadly how a sequence designated RAWMIX is in progress under automatic control and in stage 3.

If timeout alarm is obtained during execution of a step, due to the forward-stepping conditions not having been satisfied, or if the operator for some other reason wants detailed information on the sequence, he can call up an object display where the necessary information can be found.

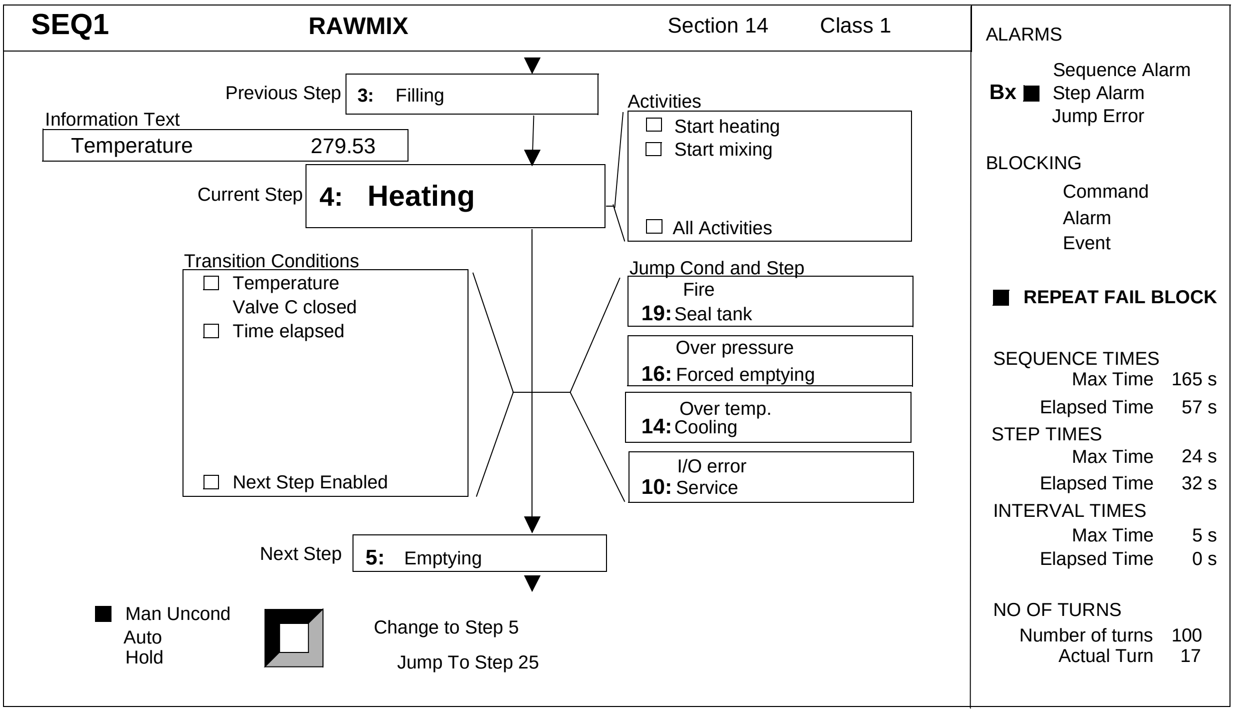


Figure 2-5. Operator’s display for a sequence control

The functional unit GROUP is designed for use with sequential start/stop of motors or sequential opening/closing of valves. The motor and valve control is assumed to be performed with the functional units MOTCON and VALVECON respectively. The text handling by GROUP in the operator’s displays is adapted to these. The sequence structure in this application variant of sequence control is also built up with the PC elements SEQ and STEP.

Up to 18 objects, motors or valves are handled as a unified group by the functional unit GROUP. To further effective the application work there are standardized type circuits with the main function defined but with certain possibilities open for project and instance adaption.

In the case of the functional unit GROUP, there are special display elements which integrate the operator’s need for information about a sequence start.

Chapter 3 SEQ

SEQ is a software package - a functional unit - designed for universal use in sequence applications.

The functional unit is standardized to a high degree to simplify the work of designing presentation, dialog and control logic. A great operational flexibility is performed by user-defined PC program where the PC elements SEQ and STEP shapes the wanted sequence structure.

3.1 Configuration

The functional unit Sequence consists of three parts:

1. A user defined PC program with the structural PC elements SEQ and STEP and an application adapted assembly of other PC elements.
2. A section for operator functions, which consists of both presentation and dialog functions. Parameters and data from the process are presented on the display screen. With a keyboard the operator can start and stop the sequence, make a manual jump, change the control mode etc.
3. A data base element with configuration constants and dynamic data which constitute an interface to the operator functions.

Figure 3-1 illustrates the relationship between these parts.

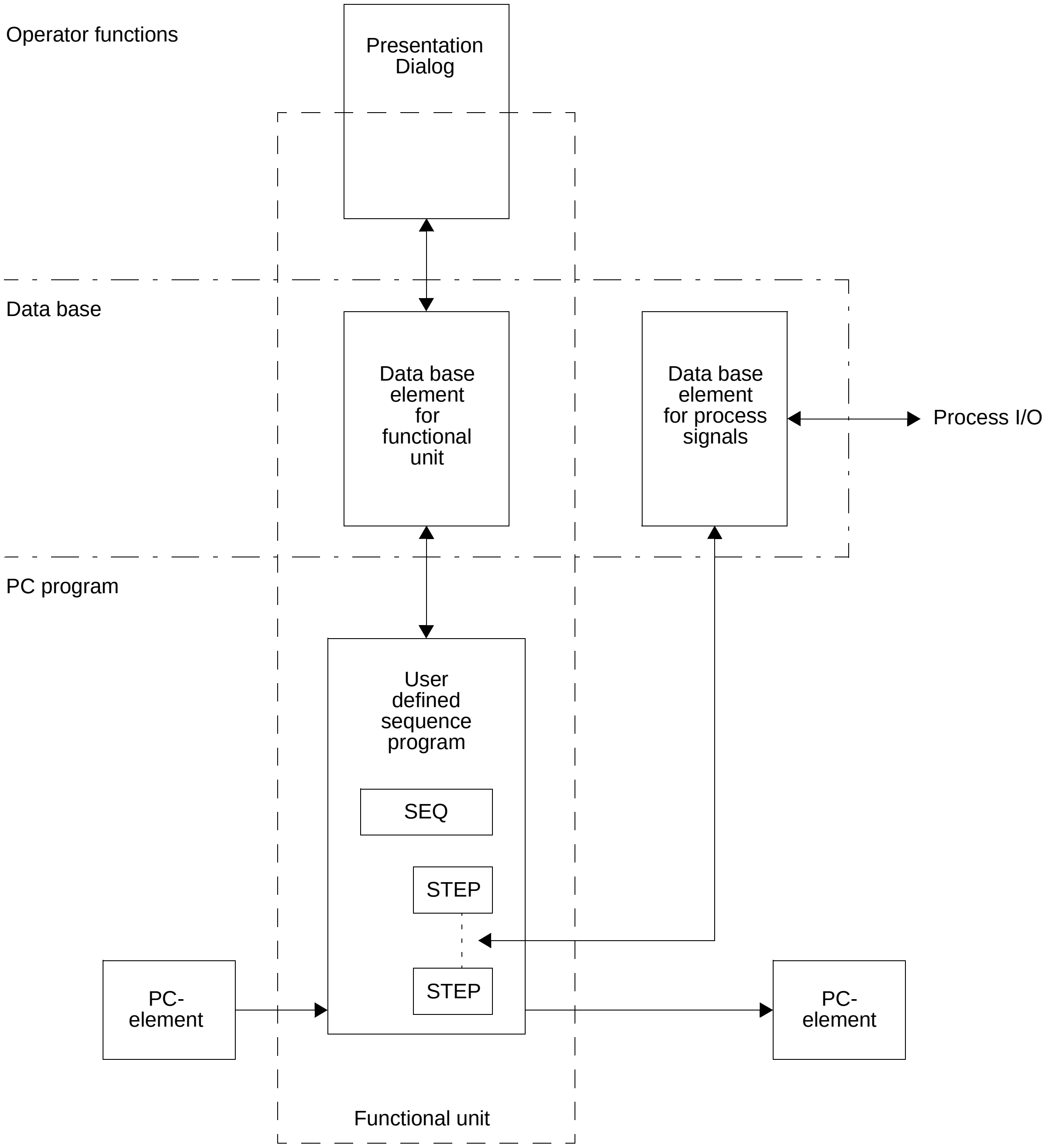


Figure 3-1. System structure for the functional unit SEQ

3.2 Function

The backbone of the Functional Unit Sequence consists of the PC elements SEQ and STEP. The sequence header SEQ is a superior execution controlling element for a number of sequence STEPs. The following description integrates the function of these main PC elements and relates to function diagrams Figure 3-2 (SEQ) and Figure 3-3 (STEP).

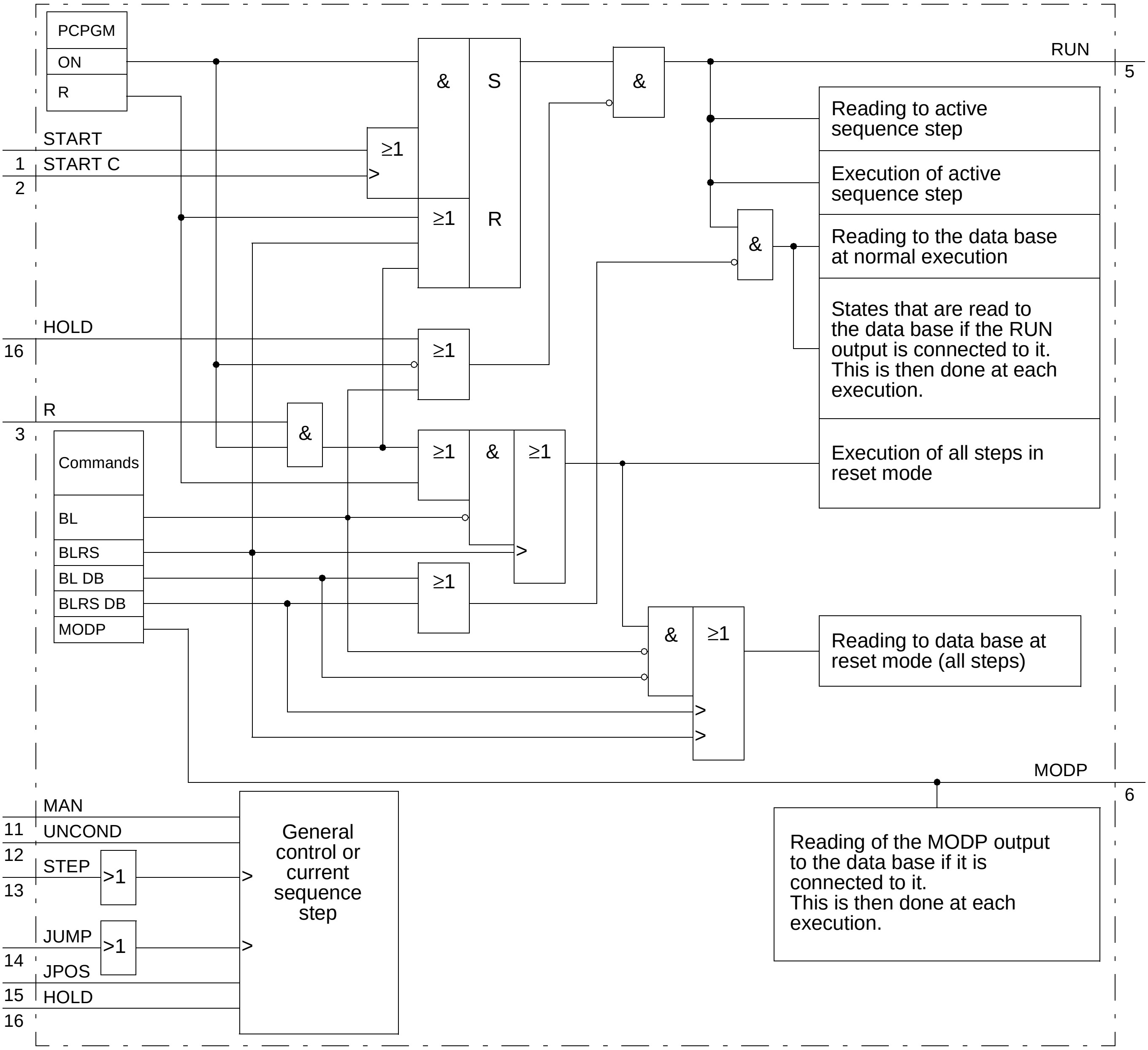

PCPGM
ON
R
START
1
START C
2
HOLD
16
R
3
Commands
BL
BLRS
BL DB
BLRS DB
MODP
MAN
11
UNCOND
12
STEP
13
>1
JUMP
>1
14
JPOS
15
HOLD
16
General control or current sequence step
&
S
≥1
R
&
RUN
5
Reading to active sequence step
Execution of active sequence step
Reading to the data base at normal execution
States that are read to the data base if the RUN output is connected to it. This is then done at each execution.
Execution of all steps in reset mode
Reading to data base at reset mode (all steps)
MODP
6
Reading of the MODP output to the data base if it is connected to it.
This is then done at each execution.

Figure 3-2. Function diagram, SEQ

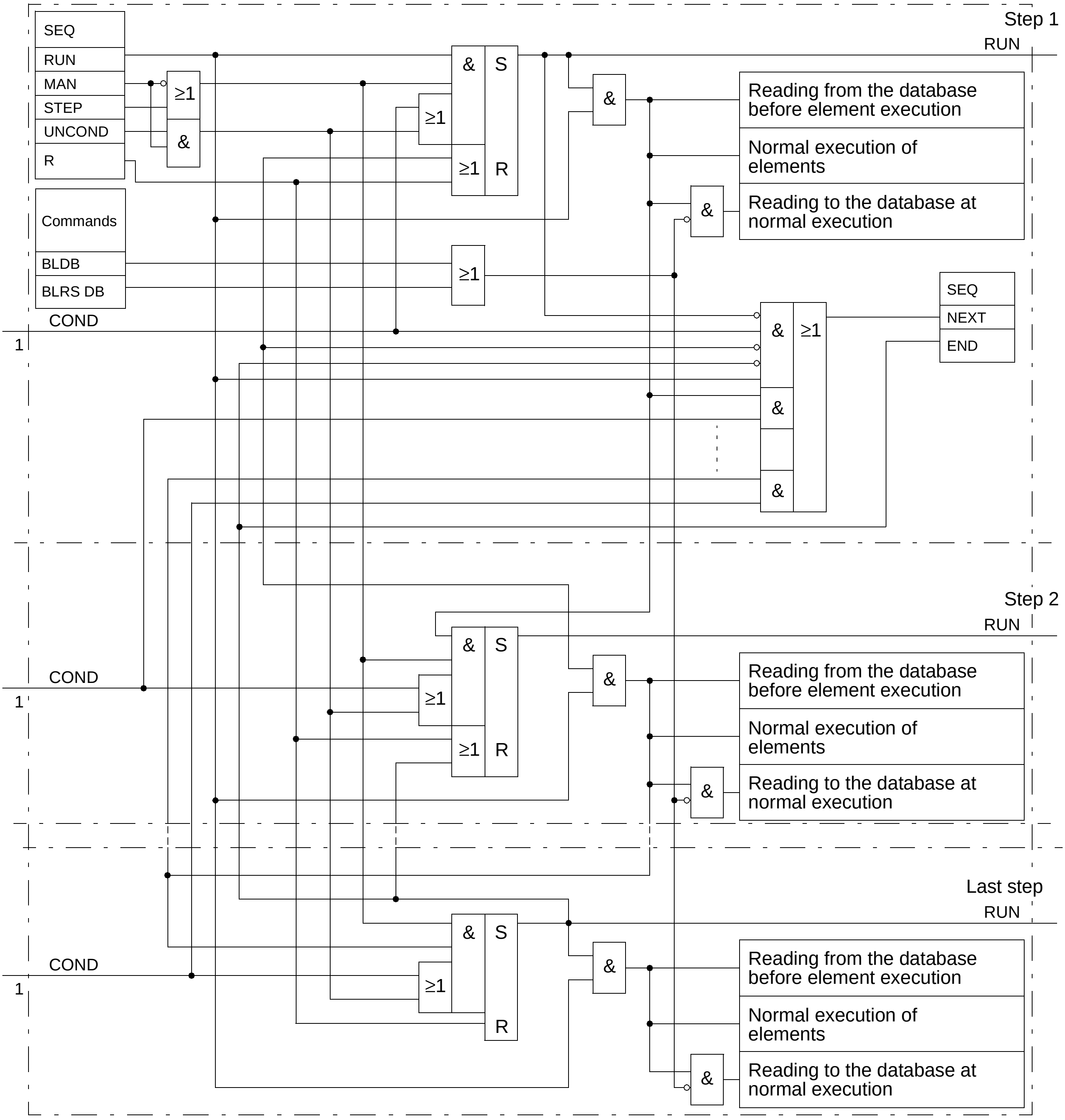

SEQ
RUN
MAN
STEP
UNCOND
R
Commands
BLDB
BLRS DB
COND
1
≥1
&
& S
≥1 R
Step 1
RUN
Reading from the database before element execution
Normal execution of elements
Reading to the database at normal execution
SEQ
NEXT
END
&
≥1
Step 2
RUN
COND
1
Reading from the database before element execution
Normal execution of elements
Reading to the database at normal execution
Last step
RUN
COND
1
Reading from the database before element execution
Normal execution of elements
Reading to the database at normal execution

Figure 3-3. Function diagram, STEP

How often the sequence is to be executed is specified with a call parameter in the sequence header SEQ. With another call parameter the place in the cycle time table is specified. The basic function in SEQ can be increased for manual control, time supervision and functions adapted to operator stations.

The step header STEP is a superior element header for separation of a sequence into steps. Function modules, blocks and all kinds of PC elements can be directly subordinate to a step header.

The element STEP has a basic function which can be extended by setting call parameters. Sections for time supervision, jumps, reset and functions for operator stations can be obtained.

3.2.1 Start of Sequence

With normal execution, the sequence can be started with a pulse at the input START of the element SEQ. The pulse is registered in an internal memory.

If a fixed 1 is connected to START, the sequence can be connected for automatic restart after resetting.

If the sequence is to be run through once, it must be started with STARTC. STARTC is a dynamic input and the internal memory for the sequence head is only affected when the input status is changed from 0 to 1.

After a start, the output RUN of the element SEQ is set.

3.2.2 Specification of Step Numbers

Each step in a sequence has a number which identifies the step. This number is used in operator station displays and for specifying jumps and alternative execution order in a sequence. The step number can be set in two different ways depending on the value of the function parameter STEPNO in the STEP elements.

1. If STEPNO=0 in the STEP elements, the step numbers are defined by the last part of the item designation of the STEP elements (e.g. 3 in PC1.1.3).
2. If STEPNO has another value than zero in the STEP elements, that value is used as the step number.

NOTE

All STEP elements must have unique (within the sequence) step numbers.

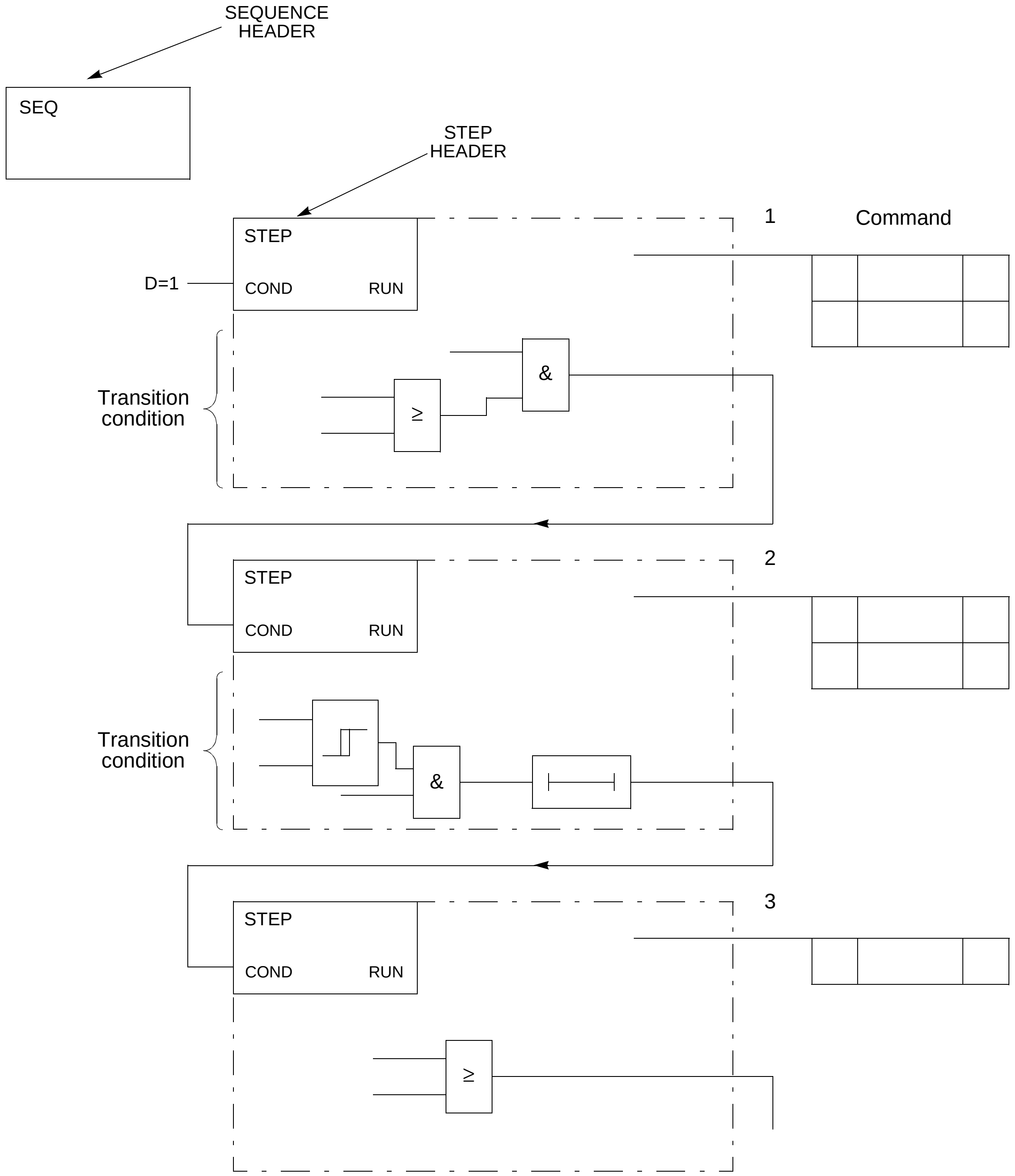


Figure 3-4. Application of single sequence

3.2.3 Normal Execution

The sequence starts if the conditions concerned at the sequence header are fulfilled and the COND input of the first STEP is 1. The first STEP and its PC elements start to execute. The RUN output of the first STEP is set, indicating an active STEP. The RUN output or any PC function, structured under the STEP element concerned can be used to give application-adapted commands to the process, e.g. stored, delayed, time-limited or pulsed commands.

Forward stepping is brought about as follows:

1. The PC program of the active step sets the COND input of the next step at 1 when the stepping conditions are satisfied (See Figure 3-4).
2. The RUN output of the active step is set to 0
3. The PC program of the active step is executed once more in the normal way
4. The RUN output of the next step is set to 1
5. The PC program of the next step is executed.

The steps 1-5 are taken sequentially at the same execution time. For this reason, external modules, e.g. CONTRM, cannot make use of output data from the sequence during stepping.

The next step is normally determined by the execution order of the sequence. However, the STEPTO terminal of the STEP element can be used to specify an alternative execution order of the sequence. In that case the next step of the sequence will be determined by the value of the STEPTO terminal instead of the normal execution order. This can for example be used for programming parallel branches in a sequence.

3.2.4 Time Functions

The sequence header SEQ as well as the step header STEP can be expanded with time functions. Selection is carried out with call parameters. For the sequence header a common signal STEPAL is also obtained which indicates any time function which has elapsed in any of the steps.

The timer is an accumulating time measurement controlled by the input ALEN. Please refer to Figure 3-5.

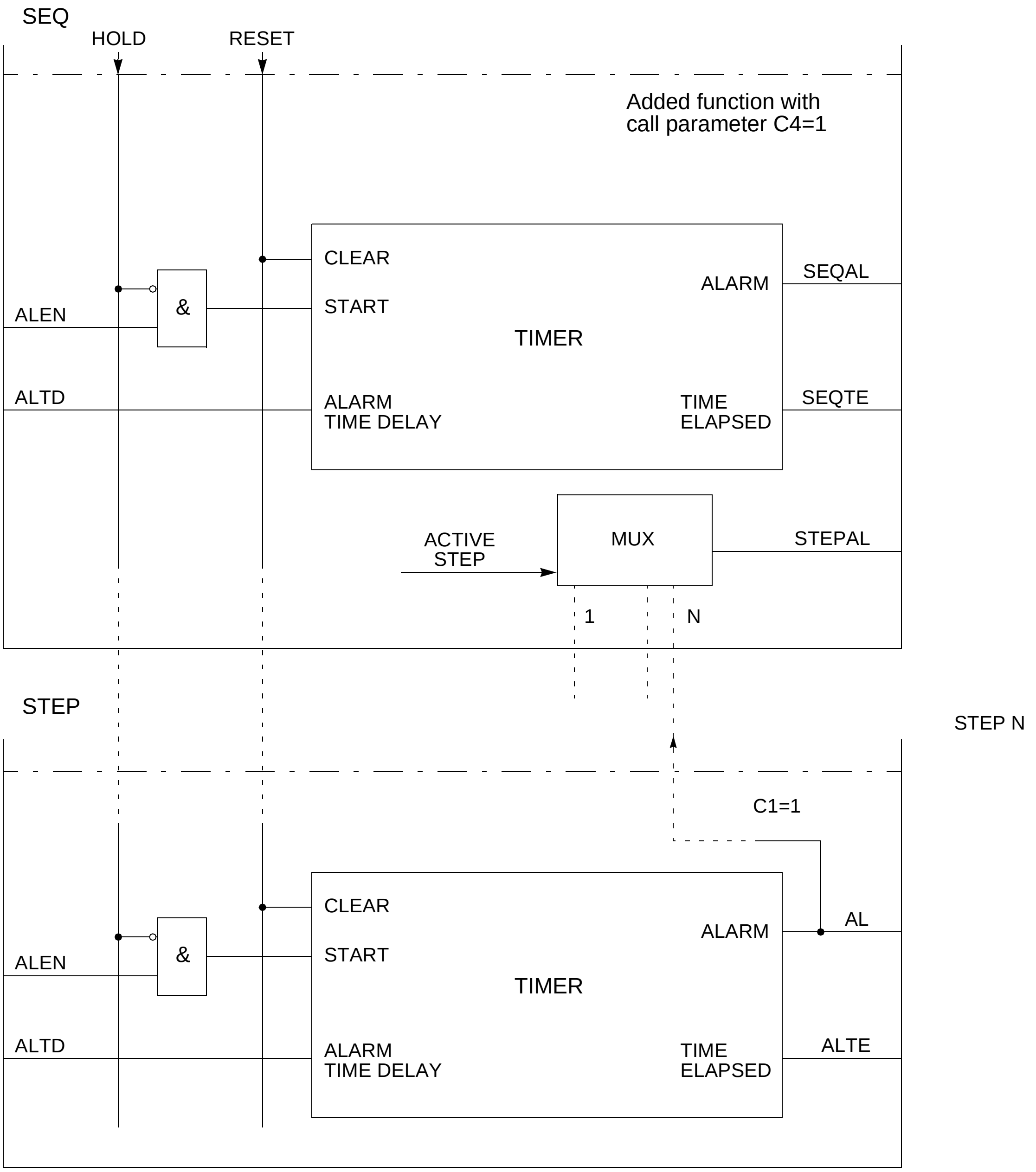


Figure 3-5. Time functions in SEQ and STEP

If ALEN is reset the timer is stopped but its contents remain and when ALEN is set, the time measurement restart from that value which applied when ALEN was reset. The setting of HOLD has the same effect on the timer as resetting ALEN. The timer can only be started when the sequence is active.

When the time specified at the input ALTD has elapsed, the output SEQAL (or AL for the step header) is set, giving alarm in the operator station.

The output SEQTE (or ALTE for the step header) gives the time which has elapsed with the timer active.

The timer is cleared when the complete sequence is cleared with the input R.

3.2.5 JUMP

3.2.5.1 Jump in Sequence

A jump can be obtained by setting the input JUMP on sequence header. The jump is made to the step which is specified at the input JPOS (jump position). For specifying step numbers, see Section 3.2.3 Normal Execution.

The jump position can be preset and the jump can be ordered from the operator station keyboard.

The jump is performed irrespective of the status at the inputs MAN, STEP, UNCOND or COND. However, if a JUMP is ordered from the operator via the dialog, the mode must be MANM.

A jump is not possible if HOLD or R are set.

3.2.5.2 Conditional Jumps

The step header STEP can be extended with conditional jump functions.

When any of the condition inputs JCOND1..JCONDC2 are set, a jump is performed to the step specified at the corresponding input of JPOS1..JPOSC2. If several condition inputs are set at the same time, a jump is made to the step specified by the input with the lowest sequential number. If the conditions for step change are satisfied at the same time as those for a jump, the jump is performed.

No jumps can be performed when the input HOLD on the superior sequence header is set. If the input MAN on the sequence header is set, a jump is not made until the input STEP is set.

When executing the step, the elements included are executed first, after which the jump conditions are checked. If both the conditions for jump and those for step change are satisfied, the jump is made.

If JPOS = 0 for the jump input concerned a jump is made to the superior sequence header. If the step specified by the JPOS input does not exist, the sequence stops.

3.2.6 HOLD

When HOLD is set on the sequence header SEQ, execution of active sequence steps is stopped. The calculated data remains until the next time the sequence step is executed. The execution is stopped and the output RUN is reset while HOLD is set to 1. The RUN output at the step header remains set.

HOLD also stops the timers in the sequence header and the step headers.

3.2.7 Manual Execution

For manual stepping of a sequence, the input MAN on the sequence header SEQ must be set to 1. The sequence is started in the same way as with automatic stepping by setting START or STARTC to 1.

When the condition for step change is satisfied, the output NEXT is set, and the step changes can be performed with a pulse at the input STEP.

If stepping is required without the condition for the next step being satisfied, the input UNCOND must be set to 1 when STEP is set to 1.

All this handling can be carried out from the operator station.

3.2.8 Clearing

3.2.8.1 Clearing of Step

When a sequence step has been active, the data calculated remains, provided that the RUN output is not included as a condition for those data.

This is utilized according to the illustrations in Figure 3-6.

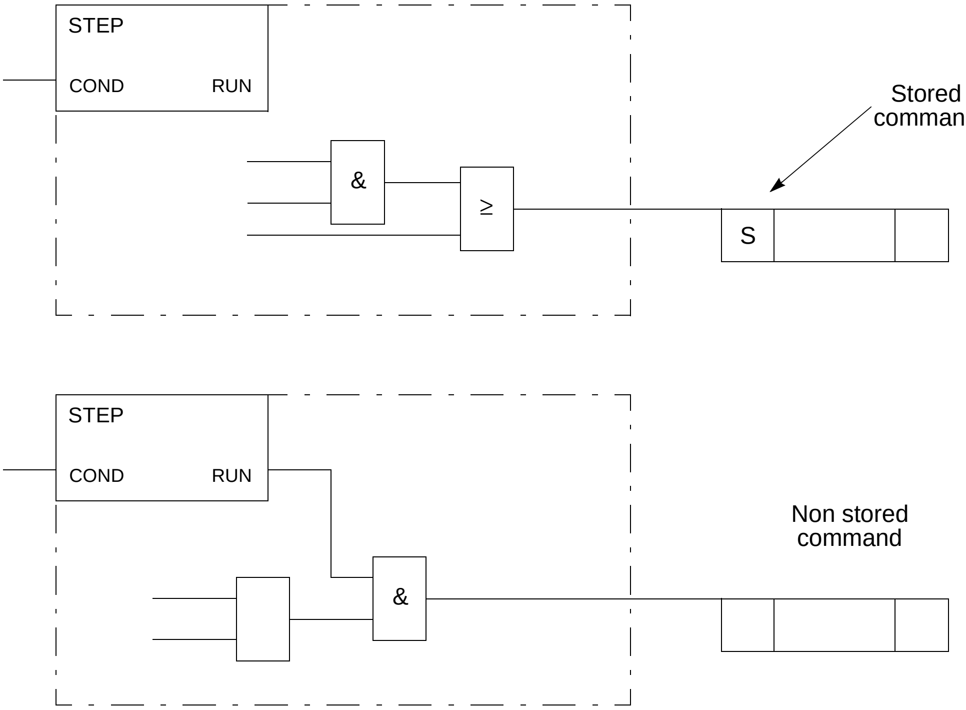


Figure 3-6. Stored, non stored commands

Calculated and stored data can be cleared by a later sequence step.

One or more steps in succession can be cleared simultaneously. The step from which clearing is to be performed is specified with an integer on one of the inputs RL1..RLC2. The steps to which clearing is to be performed is specified with an integer on the corresponding input of RH1..RHC3.

Clearing is performed when the step is exited with a normal step change or with step changing controlled by a jump condition.

If RL = 0, the complete sequence is cleared, which corresponds to a reset on the sequence header. In this case RH has no function.

If RL does not exist, there is no reset. If RH does not exist or has a value which is smaller than RL the reset is performed to the end of the sequence.

3.2.8.2 Clearing of the Sequence

If the input R on the sequence header SEQ is set, the sequence is executed in the reset mode. This means that all outputs on the elements within all of the steps of the sequence are given initial values which in most cases is the 0-value of the data. RUN is cleared by R.

The reading to the data base is performed with the data value = 0 for all data types. This means all commands to the process are reset.

For steps, there is reset execution only during the cycle when R goes to 1. For the sequence header, reset execution is performed during each cycle while R is set to 1.

Input R overrides the inputs START, STARTC and HOLD.

3.2.9 POS and END

The output POS on the sequence header SEQ specifies the step number of the sequence step which is active. If POS is 0, the sequence head is active. POS n is indicated in the operator station displays with “STEP=n”.

The output END is set when the last step in the sequence is active. This is indicated in displays with “End of sequence”.

3.2.10 Operator Functions

Sequence control built-up around the functional unit SEQ can be affected in different ways. The program designer can create possibilities of control and indication from a conventional mimic panel and/or from an operator station. The latter is described in this section.

There are standardized displays in Advant Station 500 Series Operator Station for presentation of status variations in connection with sequence controls. There are also pre-prepared dialogs for use by the operator in giving the necessary commands. A sequence control usually requires special presentation of unsatisfied transition conditions for the different steps in the sequence. With the help of such information, the operator can take the actions necessary with an unplanned stoppage of a sequence. Section 3.5 Operator Functions deals with details of these standard functions.

The following is mainly a description of how control functions in PC programs are connected to Operator functions in Advant Station 500 Series Operator Station.

The data base in Advant Controller 410/450 systems is the common storage place for command signals from and indications to the Advant Station 500 Series Operator Station. This means that the data base element is an important link in the chain of functions designated Operator Station Functions. During application programming all connections between Operator functions, PC program and process are performed via data base elements. The functional unit SEQ offers considerable flexibility for sequence control in different applications as command signals are not connected in blocks of large numbers of signals but as individual signals. This provides the program designer with freedom to enter conditions between commands in the data base element and the control in the PC element utilized. The later is principally PC element SEQ.

A diagram, Figure 3-7, illustrates the relation between dialog and terminals in the data base element which connections are to be made, as required, by the data base element and PC program. A broken line connection indicates an alternative direct connection or that an application-adapted interlocking logic can be entered. The variable content of the block connections and the standardized presentation are also presented.

NOTE

It is the connections between data base elements and PC programs which are to be performed in the PC programming. All other functional connections shown are standard functions.

For certain commands in normal applications, the continuous command signal which is obtained at the data base element is to be converted to a pulsed signal. This is done with the resetting signal UNACK_N on the PC element SEQ. UNACK_N is connected to all commands which in the application are to be automatically acknowledged. The resetting function is marked symbolically for the commands which are to be reset with the capital letter U. In practice, UNACK_N is connected directly to the terminal of the data base element e.g. = SEQ1:STEP.

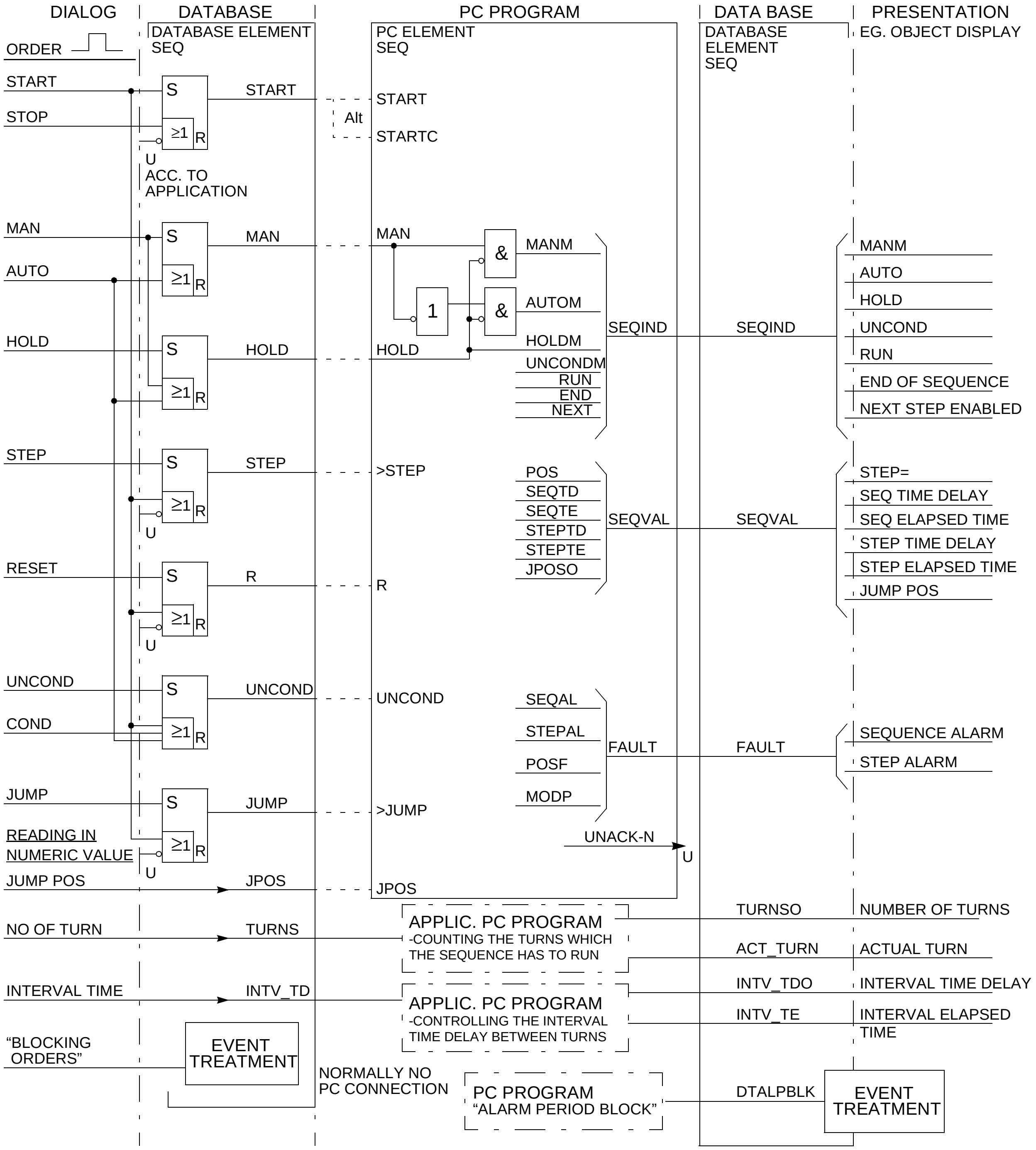


Figure 3-7. Connecting up operator functions

3.3 PC Elements

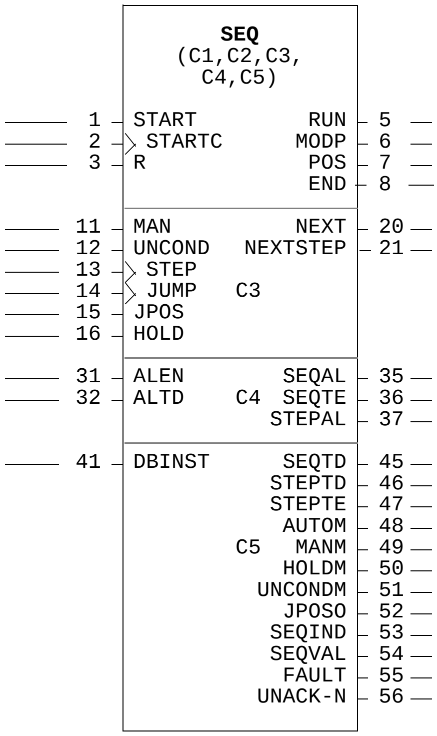
The PC elements SEQ and STEP are used for structuring the sequence program. SEQ is the general sequence controller while STEP is the controlling part of each step. Other PC elements, i.e. logical gates and time functions, are often combined with the step header. Those other PC elements define transition conditions, orders and alarm criteria.

3.3.1 PC Element SEQ

Sequence Header SEQ

Summary

SEQ is used to control a sequence.



PC element SEQ

)

Call SEQ (C1, C2, C3, C4, C5)

Parameter	Significance	Permissible values
C1	Cyclicity in ms	Permissible values are determined by the possible cycle times of the relevant PC system. These times are specified in the manual for the system concerned.
C2	Place in the cycle time table	Permissible values are determined by the relation to other execution units with the same cycle time. The maximum value is determined by the number of possible places with the same cycle time. The system selects a free place with the lowest number if no value is specified at entry.
C3	Selection of manual control	0 .. 1
C4	Selection of time function	0 .. 1
C5	Selection of MMC function	0 .. 1

Terminals Sequence Header element

No	Name	Type	Possible Operator functions(1)	Application(2)
1	START	IB	D	Input for **START** of sequence. *The sequence will start when START is true. Only a momentary signal is required. A reset is necessary to leave the end step. A continuous cycling of the sequence can be achieved with START at continuous true and a connection between END and R.*
2	STARTC	IB	D	**START C**lock. Dynamic input for start of the sequence. Used when the sequence is to be run through once only. *START Clock is a dynamic input, which will start the sequence at the leading edge. STARTC has to make a cycle after a reset, if the sequence shall start again.*
3	R	IB	D	**R**eset. Clearing of the elements of the sequence. *Reset will give the sequence initial values when true. Initial values are in most cases false and zerc for integers, reals and times. Not only the SEQ-element but also all STEP-elements will be reset.* *The reset action is performed each execution for the SEQ-element but only the first reset cycle for the STEP-elements. All values sent to the database will be 0 for all data types during reset. Reset has priority over START, STARTC and HOLD.*
5	RUN	OB	(P)	Indicates that the sequence is in operation. *RUN indicates a sequence in operation when true. RUN carries false, when R, HOLD or blocking functions from the engineering station are set true. Remark: The RUN output of the active step will remain true, if HOLD is set true, even if in this case the SEQ-terminal RUN indicates false.*

Terminals Sequence Header element

No	**Name**	**Type**	**Possible Operator functions**[1]	**Application**[2]
6	MODP	OB	(P)	**MOD**ify **P**ermission. Specifies if the sequence may be modified from the engineering station. *MODify Permission indicates when true, that a modificatson mode of the engineering station is set by the application engineer for this sequence. For further information see documentation on Advant Station 100 Series Engineering Station.*
7	POS	OI	(P)	**POS**ition. Output which specifies the step number of the active step. *POSition indicates the number of the active step. The position 0 indicates, that the only active part is the sequence header SEQ. POS is displayed as STEP in the Operator Station.*
8	END	OB	(P)	Output which is active when the last step of the sequence is active. *END indicates when true, that the last step in the sequence is active.*
11	MAN	IB	D	**MAN**ual. Input which is to be set for manual stepping of the sequence. *MANual is the manual mode selection input. START or STARTC is still required to start the sequence.*
12	UNCOND	IB	D	**UNCOND**itional. Input which permits stepping even when the condition for the next step is not satisfied. *UNCONDitional releases the possibility to step without fulfilled step criteria.*
13	STEP	IB	D	Input for **STEP** change. *STEP is a dynamic input, which will step the sequence one step. MAN and UNCOND or the COND at the step must be true for a step action.*
14	JUMP	IB	D	Input for **JUMP** to step specified by JPOS. *JUMP is a dynamic input, which activates a jump in the sequence. HOLD and R are blocking jumps. A JUMP is carried out independent of UNCOND or COND at the step.* *Remark: If JUMP is ordered from another PC element, the jump will be performed independent on MAN or AUTO. However, if a JUMP is ordered from the operator via the dialog, the mode must be MAN.*
15	JPOS	II	D	**J**ump **POS**ition. Input which specifies the last part of the record designation for the step to which the jump is to be made. *Jump POSition specifies the step to which the sequence has to jump. JPOS = 0 is the sequence header.*
16	HOLD	IB	D	Input which is used to stop the execution. *HOLD stops further execution of the active step, but still keeping the RUN output of the step true. The RUN output of the sequence will however turn to false. All time supervisions will stop their timers.*

Terminals Sequence Header element

No	**Name**	**Type**	**Possible Operator functions**[1]	**Application**[2]
20	NEXT	OB	(P)	Output which is set when any condition for step change or jump is satisfied or when UNCOND is set. *NEXT indicates, when true, that the condition for the next step, is fulfilled or UNCOND is set true.*
21	NEXTSTEP	OI	(P)	Step to which stepping in manual mode is performed. Is connected to the corresponding input of the data base element when MMC function is used. *The NEXTSTEP when in manual mode of control. Should be connected to the correspondingly named input on the database element when the MMC function is used.*
31	ALEN	IB	–	**AL**arm **EN**able. Input for start and stop of time measurement. *ALarm ENable is the time supervision controlling input. ALEN set to false will stop the sequence timer but still store its value.*
32	ALTD	IT	–	**AL**arm **T**ime **D**elay. Input for time after which SEQAL is to be set. *ALarm Time Delay is the time input defining the time frame before alarm - SEQAL - of the I sequence total time.*
35	SEQUAL	OB	(P)	**SEQ**uence **AL**arm. Output which is set when the time specified on ALTD has elapsed. *SEQuence ALarm indicates when true, that the time defined by ALTD has elapsed.*
36	SEQTE	OT	(P)	**SEQ**uence **T**ime **E**lapsed. Output for elapsed time. *SEQuence Time Elapsed indicates the elapsed t time of the sequence time supervision ALTD.*
37	STEPAL	OB	(P)	**STEP AL**arm. Output for summary signal which indicates that the time set in an active subordinate step has elapsed. *STEP ALarm is a common signal, which when true indicates that the active step has elapsed time.*
41	DBINST	II	–	**D**ata **B**ase **INST**ance. The instance number of the sequence in the data base index, alternatively the name of the instance. *Data Base INSTance is the identification and reference terminal to the database. The following formats are possible.*
45	SEQTD	OT	(P)	**SEQ**uence **T**ime **D**elay. Output that indicates ALTD. *SEQuence Time Delay indicates the time set at the terminal ALTD of the SEQ-element.*
46	STEPTD	OT	(P)	**STEP T**ime **D**elay. Output giving ALTD for the active step. *STEP Time Delay indicates the time set at the terminal ALTD of the active STEP-element.*

Terminals Sequence Header element

No	**Name**	**Type**	**Possible Operator functions**[1]	**Application**[2]
47	STEPTE	OT	(P)	**STEP T**ime **E**lapsed. Output for time that has elapsed in the timer of the active step. *STEP Time Elapsed indicates the time at the ALTE output of the active STEP-element.*
48	AUTOM	OB	(P)	**AUTO M**ode. Output which is set when operating in mode AUTO. *AUTO Mode indicates if the sequence is in automatic mode. This occur when MAN is set to false.*
49	MANM	OB	(P)	**MAN**ual **M**ode. Output which is set when operating in mode MAN. *MANual Mode indicates if the sequence is in manual mode set by the MAN input.*
50	HOLDM	OB	(P)	**HOLD M**ode. Output which is set when operating in mode MAN. *HOLD Mode indicates if the sequence is in hold mode set by the HOLD input.*
51	UNCONDM	OB	(P)	**UNCOND**itional **M**ode. Output which is set when operating in mode MAN with UNCOND. *UNCONDitional Mode indicates the status of the input UNCOND.*
52	JPOSO	OI	(P)	**J**ump **POS**ition **O**utput. Output that indicates the jump target JPOS. *Jump POSition Output indicates the value at JPOS.*
53	SEQIND	OI	P	**SEQ**uence **IND**ications. Output for status information to the data base element. *SEQuence INDication is a block terminal, which has to be connected to the data base element for the transfer of data to the MMC-function. The following variables are included: AUTOM, MANM, HOLDM, UNCONDM, RUN, END, NEXT.*
54	SEQVAL	OA20	P	**SEQ**uence **VAL**ues. Output for value information to the data base element. *SEQuence VALues is a block terminal, which has to be connected to the data base element for the transfer of data to the MMC-function. The following variables are included: POS, SEQTD, SEQTE, STEPTD, STEPTE, JPOSO.*

Terminals Sequence Header element

No	**Name**	**Type**	**Possible Operator functions**[1]	**Application**[2]
55	FAULT	OI	P	Output for error information to the data base element. *FAULT is a block terminal, which has to be connected to the data base element for the transfer of data to the MMC-function.* *The following variables are included: SEQAL, STEPAL, POSF, MODP.* *The variable POSF is an internal variable in the PC element SEQ not available at an individual terminal. POSF is set when a jump is ordered and the step, which is defined by JPOS does not exist.*
56	UNACK-N	OB	–	**UNACK**nowledged**-N**. Output to the data base element for clearing of dialogue. *UNACKnowledge - Negative is an output, whose purpose is to reset operator orders in the data base element. The output is set false at each end of a sequence execution. Those data base terminals connected with the UNACK-N will the be reset. The achievement is a transfer of an operator order from a continuous signal to a momentary, e.g. Reset. For further information see Section 3.2.10 Operator Functions or Section 3.6 Application Examples.*

(1) The information in the column “Possible Operator Functions” is unique for this manual. It is not included in the manual *PC Elements Advant Controller 400 Series Reference Manual*.

(2) The texts in italic in the column “Application” are unique for this manual. They are not included in the manual *PC Elements Advant Controller 400 Series Reference Manual*.

3.3.1.1 Connections to Operator Station

The functional unit SEQ offers more Operator functions than can be utilized by the PC element SEQ. A summary of all possibilities is given in Section 3.2.10 Operator Functions. The section also contains a documentation of the cross-connections necessary between PC program and data base element in the functional unit SEQ.

The PC element SEQ is responsible for a very large proportion of the possible Operator functions. In the table Terminals in the preceding section, Section 3.3.1 PC Element SEQ, the connections on the PC element SEQ which have a prepared Operator function, are included in the column **Possible Operator functions**.

D (Dial) marks that a dialog function is available in the standard dialog of the functional unit in the Advant Station 500 Series Operator Station.

P (Presentation) marks that a presentation function is available in standard displays in the Advant Station 500 Series Operator Station.

These connections can be connected to the data base element SEQ in the application program if the function is required.

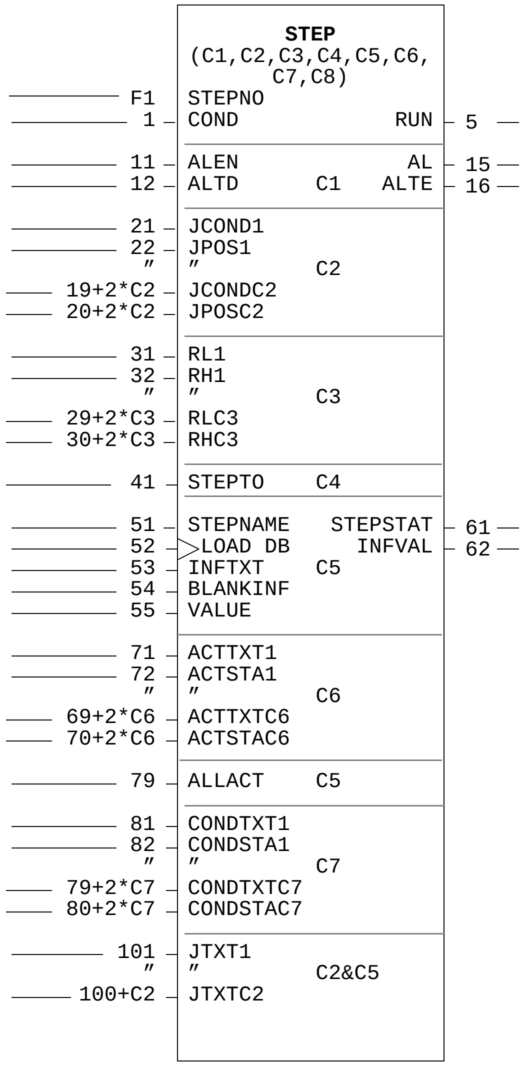
(P) indicates that the connection concerned is not made to the data base element as an individual signal but that the connection is included in one of the block connections SEQIND, SEQVAL or FAULT.

3.3.2 PC element STEP

Step Header STEP

Summary

The step header STEP is used to partition a sequence into sequence steps.



PC element STEP

Call STEP (C1, C2, C3, C4, C5, C6, C7, C8)

Parameter	Significance	Permissible values
C1	Selection of time function	0..1
C2	Number of jumps	0..4

Call STEP (C1, C2, C3, C4, C5, C6, C7, C8)

Parameter	Significance	Permissible values
C3	Number of reset groups	0..4
C4	Selection of alternative execution order	0..1
C5	Selection of MMC function	0..1
C6	Number of activities	0..4
C7	Number of step conditions	0..8
C8	Maximum number of characters in the texts.	0..15
If C5=0 then must C6=C7=C8=0 If C5=1 then must C8>0		

Terminal Description

No	Name	Type	Description
F1	STEPNO	FI	**STEP N**umber. Defines the step number of the step element. If STEPNO is anything but zero, STEPNO is used as the step number. If STEPNO is zero, the last part of the item designation of the STEP element is used as the step number. The STEP number is not used for normal sequence execution, only as input to alternative execution order (e.g. as input to the JPOS and STEPTO terminals). Range: 0..999.
1	COND	IB	**COND**ition. Condition input for stepping.
5	RUN	OB	Indicates that the step is active.
11	ALEN	IB	**AL**arm **EN**able. Input for start and stop of time measurement.
12	ALTD	IT	**AL**arm **T**ime **D**elay. Input for alarm time.
15	AL	OB	**AL**arm. Output for alarm. Set (to 1) when the time specified at the input ALTD has elapsed.
16	ALTE	OT	**AL**arm **T**ime **E**lapsed. Output for elapsed time.
21	JCOND1	IB	**J**ump **COND**ition **1**. Input for jump to step specified by JPOS1.
22	JPOS1	II	**J**ump **POS**ition **1**. Specifies to which step the jump is to be made if JCOND1 is set.
23	JCOND2	IB	**J**ump **COND**ition **2**. Input for jump to step specified by JPOS2.
24	JPOS2	II	**J**ump **POS**ition **2**. Specifies to which step the jump is to be made if JCOND2 is set.
: 19+2xC2	JCOND*C2*	IB	**J**ump **COND**ition ***C2***. Input for jump to step specified by JPOS*C2*.
20+2xC2	JPOS*C2*	IB	**J**ump **POS**ition ***C2***. Specifies to which step the jump is to be made if JCOND*C2* is set.
31	RL1	II	**R**eset **L**ow limit **1**. Input for the number of the first step in the first group of steps to be cleared.
32	RH1	II	**R**eset **H**igh limit **1**. Input for the number of the last step in the first group of steps to be cleared.
33	RL2	II	**R**eset **L**ow limit **2**. Input for the number of the first step in the second group of steps to be cleared.

Terminal Description

No	Name	Type	Description
34	RH2	II	**R**eset **H**igh limit **2**. Input for the number of the last step in the second group of steps to be cleared.
: 29+2xC3	RL*C3*	II	**R**eset **L**ow limit ***C3***. Input for the number of the first step in the last group of steps to be cleared.
30+2xC3	RH*C3*	II	**R**eset **H**igh limit ***C3***. Input for the number of the last step in the last group of steps to be cleared.
41	STEPTO	II	**STEP T**O. This input modifies the execution order of the sequence. The input determines which step should be next step in the sequence. To be used in convergence of a sequence selection.
51	STEPNAME	IA12	**STEP NAME**. Name of the sequence step. Max 12 characters.
52	LOAD_DB	IB	**LOAD D**ata**B**ase. Dynamic input that updates the data base with all texts in the element. This input should be used only if you want to change some of the texts during execution of the step.
53	INFTXT	IAC8	**INFO**rmation **T**e**XT**. Information text for the sequence step.
54	BLANKINF	IB	**BLANK INF**ormation text. Input set if no information text is to be displayed.
55	VALUE	IR	Information **VALUE**. Input for value to Operator Station.
61	STEPSTAT	OIL	**STEP STATUS** output. Connection to the database element.
62	INFVAL	OR	Value output. Connection to the database element.
71	ACTTXT1	IAC8	**ACT**ivity **T**e**XT 1**. Text describing activity 1.
72	ACTSTA1	IB	ACTivity STAtus 1. Input determining status of activity 1.
69+2xC6	ACTTXTC6	IAC8	**ACT**ivity **T**e**XT C6**. Text describing activity C6 in the sequence step.
70+2xC6	ACTSTAC6	IB	**ACT**ivity **STA**tus **C6**. Input determining status of activity C6.
79	ALLACT	IB	**ALL ACT**ivities. Input confirming that all activities in the sequence step has been executed.
81	CONDTXT1	IAC8	**COND**ition **T**e**XT 1**. Text describing step condition 1.
82	CONDSTA1	IB	**COND**ition **STA**tus **1**. Input determining status of step condition 1.
: 79+2xC7	CONDTXTC7	IAC8	**COND**ition **T**e**XT C7**. Text describing step condition C7.
80+2xC7	CONDSTAC7	IB	**COND**ition **STA**tus **C7**. Input determining status of step condition C7.
101	JTXT1	IAC8	**J**ump **T**e**XT 1**. Text describing jump condition 1.
: 100+C2	JTXTC2	IAC8	**J**ump **T**e**XT C2**. Text describing jump condition C2.

3.4 Data Base Element SEQ

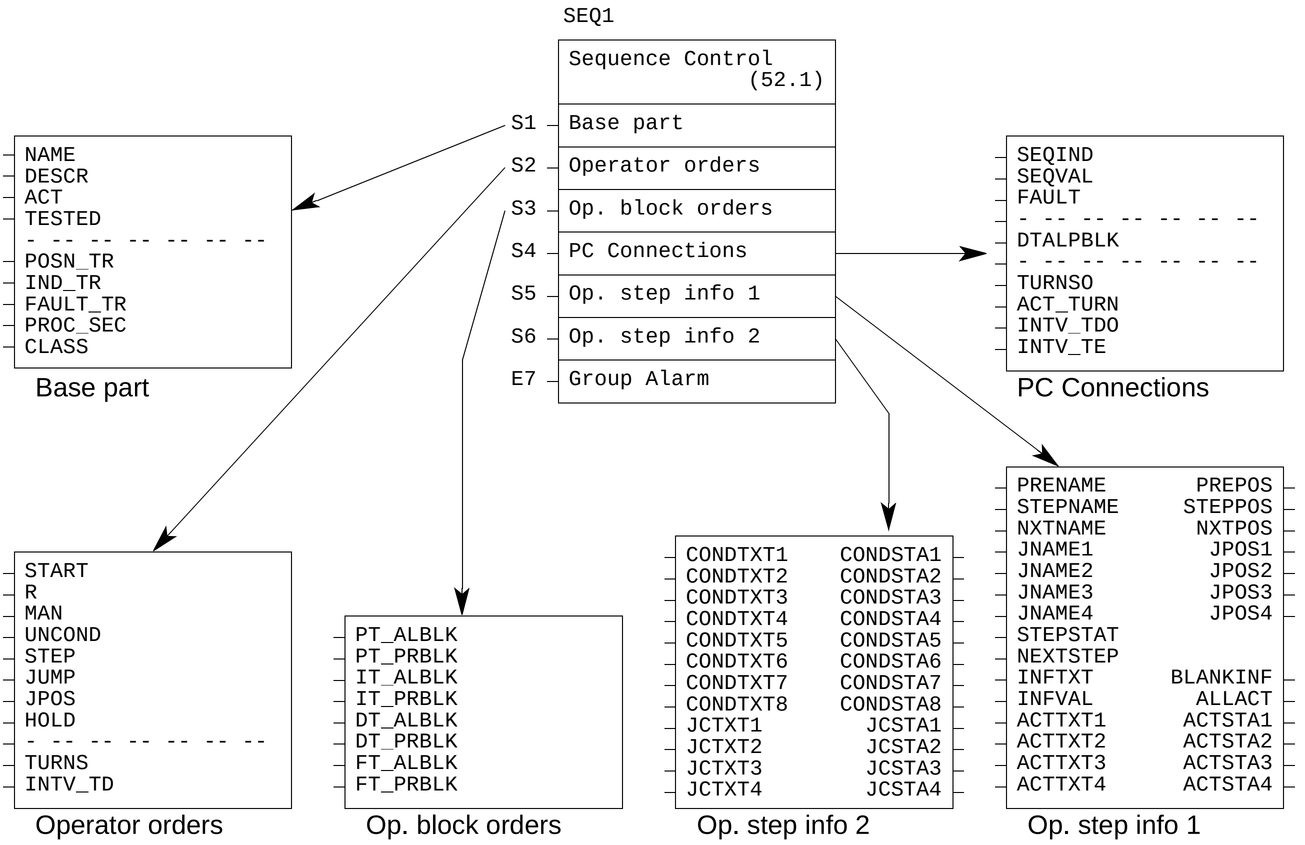
Sequence Control SEQ

Summary

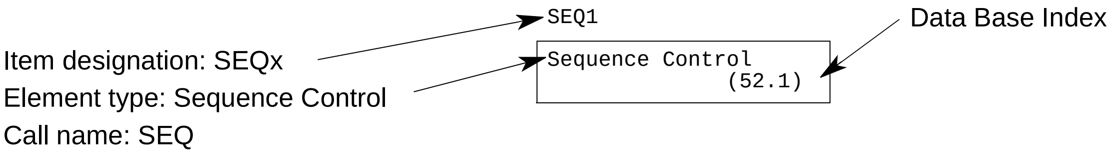
The Sequence Control data base element provides the interface between Advant Controller 400 Series and Advant Station 500 Series Operator Station to enable presentation and control from the Advant Station 500 Series Operator Station operating terminal.

For information about the extension parts of Group Alarm see data base element Group Member.

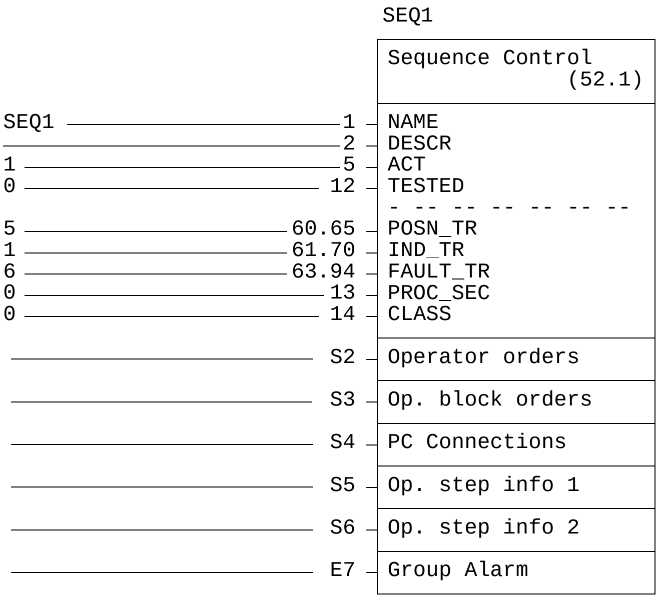
Overview



Head



Base part



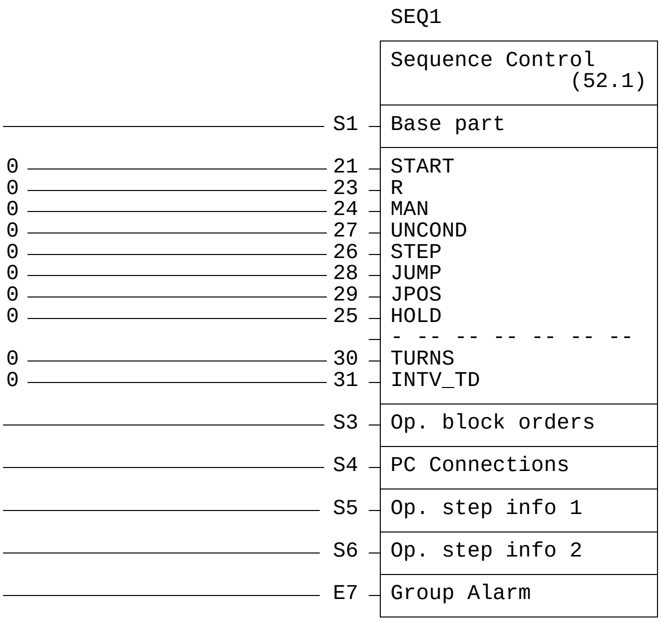
Terminal Description, Base part

Terminal No	Terminal Name	Value entered by	Default value	PC connection data type	Description	Remarks
1	NAME	user	SEQx	–	Each sequence must have a unique **NAME**.	Max. 20 characters. Spaces are not allowed.
2	DESCR	user	–	–	This **DESCR**iption of the sequence will be used in pictures and lists in Operator Station.	Max. 20 characters. You may enter any information regarding the sequence.
5	ACT	user	1	B(r/w)	1=the element is **ACT**ive.	–
12	TESTED	user	0	B(r/w)	Can be set by the user to indicate that the element has been **TESTED** at start-up.	Does not affect the system.

Terminal Description, Base part (Continued)

Termi-nal No	Terminal Name	Value entered by	Default value	PC con-nection data type	Description	Remarks
60.65	POSN_TR	user	5	–	**POS**itio**N TR**eatment defines the event handling in Operator Station of position changes.	Changes in the following variable is included: POS.
61.70	IND_TR	user	1	–	**IND**ication **TR**eatment defines the event handling in Operator Station of status changes.	Changes in the following variables are included: RUN, END, AUTOM, MANM, HOLDM, UNCONDM, NEXT.
63.94	FAULT_TR	user	6	–	**FAULT TR**eatment defines the event handling in Operator Station of individual faults.	Changes in the following variables are included: SEQAL, STEPAL, POSF.
13	PROC_SEC	user	0	I(r)	**PROC**ess **SEC**tion is used for sectioning the alarm handling. Section numbers 1-16.	0=no sectioning. The signal may be presented in lists –1=the signal cannot be selected and cannot be presented in lists.
14	CLASS	user	0	I(r)	**CLASS** subdivides process section. Used by status list function in Operator Station	–

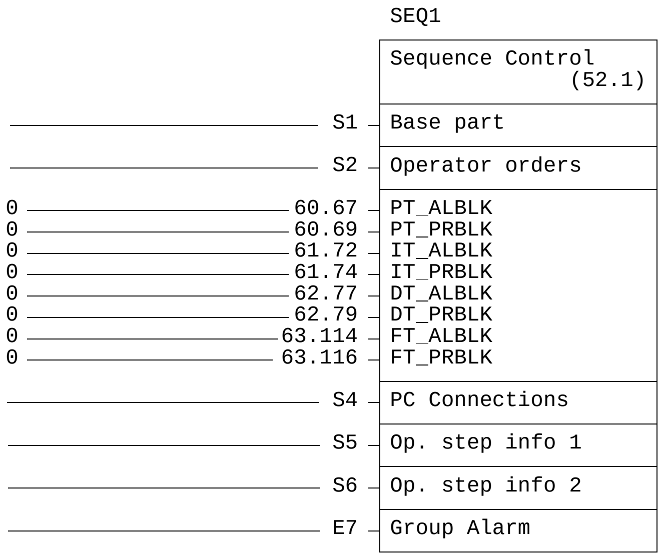
Operator orders



Terminal Description, Operator orders

Terminal No	**Terminal Name**	**Value entered by**	**Default value**	**PC connection data type**	**Description**	**Remarks**
21	START	dialog	0	B(r/w)	**START**s the sequence. A reset is necessary to restart.	
23	R	dialog	0	B(r/w)	**R**eset gives the sequence initial values and resets the sequence.	–
24	MAN	dialog	0	B(r/w)	**MAN** sets the sequence in manual mode.	–
27	UNCOND	dialog	0	B(r/w)	**UNCOND**itional, release the possibility to step without fulfilled step criterias.	–
26	STEP	dialog	0	B(r/w)	**STEP**s the sequence one step.	–
28	JUMP	dialog	0	B(r/w)	**JUMP** changes the active step to the step pointed out by JPOS.	–
29	JPOS	dialog	0	I(r/w)	**J**ump **POS**ition specifies the step, which will be active on a JUMP.	–
25	HOLD	dialog	0	B(r/w)	**HOLD** interlocks further stepping and the sequence remains in present status, with timers stopped.	–
30	TURNS	dialog	0	I(r/w)	Number of **TURNS**, which the sequence has to run.	Turns must be used by a PC element counting the turns, if the terminal shall have any influence on the system.
31	INTV_TD	dialog	0	IL(r/w)	**INT**er**V**al **T**ime **D**elay between turns.	INTV_TD must be used by a PC element controlling the time, if the terminal shall have any influence on the system.

Op. block orders



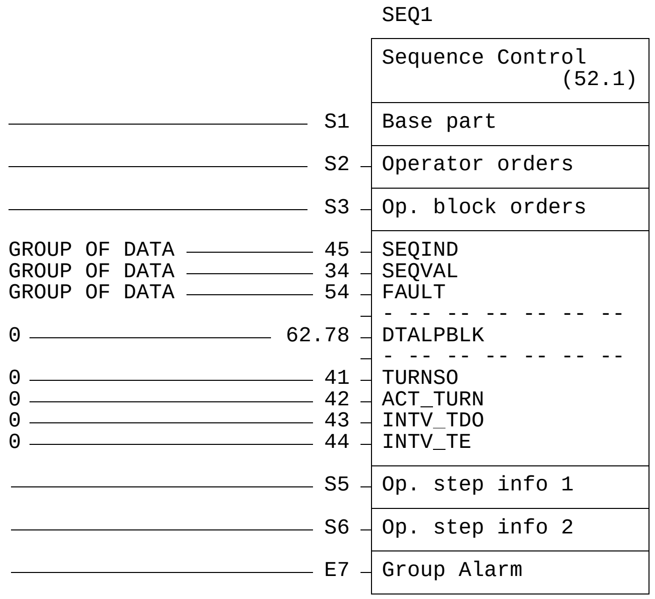
Terminal Description, Op. block orders

Terminal No	Terminal Name	Value entered by	Default value	PC connection data type	Description	Remarks
60.67	PT_ALBLK	dialog	0	–	**P**osition **T**reatment_**AL**arm **BL**oc**K**, blocks the up-dating of OS files including events caused by position changes.	–
60.69	PT_PRBLK	dialog	0	–	**P**osition **T**reatment_**PR**int **BL**oc**K**, blocks the printing of events caused by position changes.	–
61.72	IT_ALBLK	dialog	0	–	**I**ndication **T**reatment_**AL**arm **BL**oc**K**, blocks the up-dating of OS files including events caused by status changes.	–
61.74	IT_PRBLK	dialog	0	–	**I**ndication **T**reatment_**PR**int **BL**ocK, blocks the printing of events caused by status changes.	–
62.77	DT_ALBLK	dialog	0	–	**D**isturbance **T**reatment_**AL**arm **BL**oc**K**, blocks the up-dating of OS files including alarms caused by disturbance changes.	–

Terminal Description, Op. block orders (Continued)

Terminal No	Terminal Name	Value entered by	Default value	PC connection data type	Description	Remarks
62.79	DT_PRBLK	dialog	0	–	**D**isturbance **T**reatment_**PR**int **BL**oc**K**, blocks the printing of alarms caused by disturbance and operator order changes.	–
63.114	FT_ALBLK	dialog	0	–	**F**ault **T**reatment_**AL**arm **BL**oc**K**, blocks the up-dating of OS files including the alarms caused by changes of single errors.	–
63.116	FT_PRBLK	dialog	0	–	**F**ault **T**reatment_**PR**int **BL**oc**K**, blocks the printing of alarms caused by changes of single errors.	–

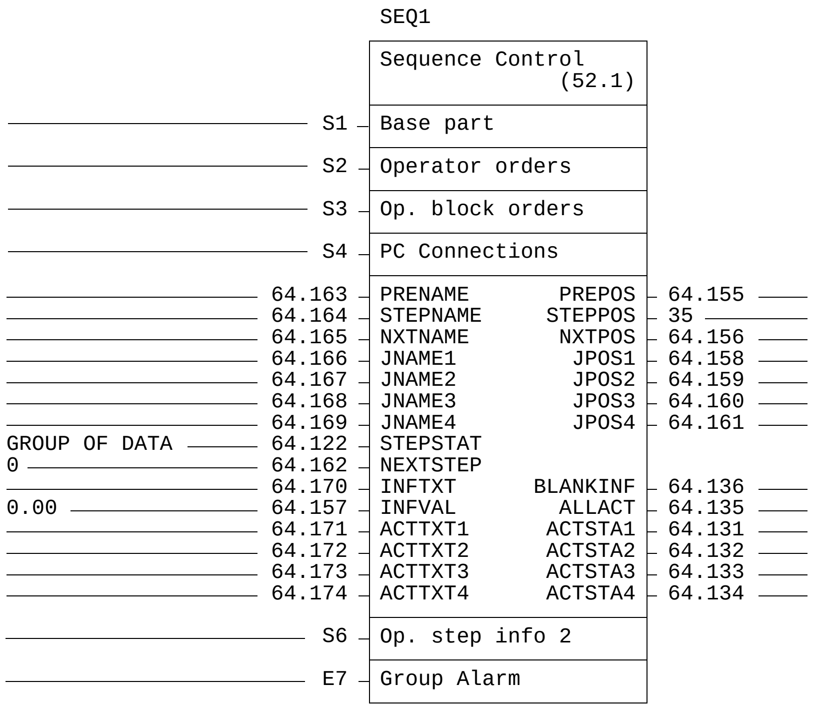
PC Connections



Terminal Description, PC Connections

Terminal No	Terminal Name	Value entered by	Default value	PC connection data type	Description	Remarks
45	SEQIND	system	–	I(r/w)	**SEQ**uence **IND**ication is a block terminal carrying indication information.	The following variables are included: AUTOM, MANM, HOLDM, UNCONDM, RUN, END, NEXT. This terminal has to be referred by the SEQ PC element, output SEQIND.
34	SEQVAL	system	–	A20(r/w)	**SEQ**uence **VAL**ues is a block terminal carrying value information.	The following variables are included: POS, SEQTD, SEQTE, STEPTD, STEPTE, JPOSO. This terminal has to be referred by the SEQ PC element, output SEQVAL.
54	FAULT	system	–	I(r/w)	**FAULT** is a block terminal carrying type error information.	The following variables are included: SEQAL, STEPAL, POSF. This terminal has to be referred by the SEQ PC element, output FAULT.
62.78	DTALPBLK	system	–	B(r/w)	**D**isturbance **T**reatment **AL**arm **P**eriod **BL**oc**K**, blocks updating and printout in Operator Station, including summation errors.	The terminal can be referred by any PC element, which then has the blocking power.
41	TURNSO	system	–	I(r/w)	**TURNS O**utput is the indication terminal of turns.	The terminal is referred by the PC element counting the turns, if the function is utilized.
42	ACT_TURN	system	–	I(r/w)	**ACT**ual**_TURN** is the indicating terminal of the actual turn.	The terminal is referred by the PC element counting the turns, if the function is utilized.
43	INTV_TDO	system	–	IL(r/w)	**INT**er**V**al **T**ime **D**elay **O**utput is the indicating terminal of interval time delay.	The terminal is referred by the PC element controlling the interval time, if the function is utilized.
44	INTV_TE	system	–	IL(r/w)	**INT**er**V**al **T**ime **E**lapsed is the indicating terminal of interval time elapsed.	The terminal is referred from the PC element counting the interval time if the feature is utilized.

Op. step info 1


SEQ1
Sequence Control (52.1)
S1 Base part
S2 Operator orders
S3 Op. block orders
S4 PC Connections
64.163 PRENAME PREPOS 64.155
64.164 STEPNAME STEPPOS 35
64.165 NXTNAME NXTPOS 64.156
64.166 JNAME1 JPOS1 64.158
64.167 JNAME2 JPOS2 64.159
64.168 JNAME3 JPOS3 64.160
64.169 JNAME4 JPOS4 64.161
GROUP OF DATA 64.122 STEPSTAT
0 64.162 NEXTSTEP
64.170 INFTXT BLANKINF 64.136
0.00 64.157 INFVAL ALLACT 64.135
64.171 ACTTXT1 ACTSTA1 64.131
64.172 ACTTXT2 ACTSTA2 64.132
64.173 ACTTXT3 ACTSTA3 64.133
64.174 ACTTXT4 ACTSTA4 64.134
S6 Op. step info 2
E7 Group Alarm

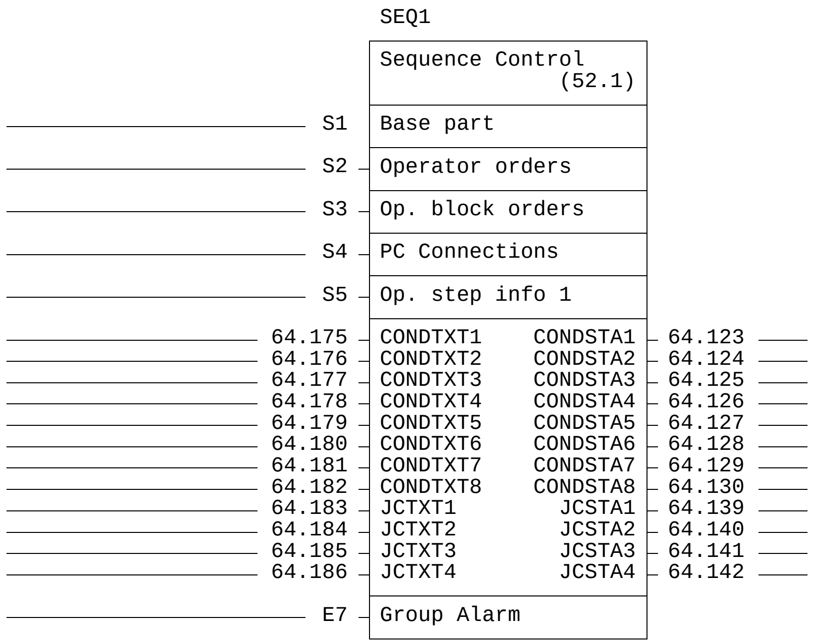
Terminal Description, Op. step info 1

Terminal No	Terminal Name	Value entered by	Default value	PC connection data type	Description	Remarks
64.163	PRENAME	system	–	–	**PRE**vious step **NAME**.	–
64.164	STEPNAME	system	–	–	Current **STEP NAME**.	–
64.165	NXTNAME	system	–	–	**N**e**XT** step **NAME**.	–
64.166 : 64.169	JNAMEx	system	–	–	**J**ump **NAME x**. For x=1..4. Name of the step to which a jump will take place if condition x is fulfilled.	–
64.122	STEPSTAT	PC	–	IL(r/w)	**STEP STAT**us. To be connected to the PC element STEP.	–
64.162	NEXTSTEP	PC	0	IL(r/w)	**NEXT STEP**. No of next step by manual mode jumping. A zero value means that no jump condition is fulfilled.	–

Terminal Description, Op. step info 1 (Continued)

Terminal No	Terminal Name	Value entered by	Default value	PC connection data type	Description	Remarks
64.170	INFTXT	system	–	–	**INF**ormation **T**e**XT**.	–
64.157	INFVAL	PC	0.00	R(r/w)	**INF**ormation **VAL**ue. Value associated with INFTXT.	Two decimals displaying.
64.171 : 64.174	ACTTXTx	system	–	–	**ACT**ivity **T**e**XT x**. For x=1..4. Text describing activity x for current step.	–
64.155	PREPOS	system	–	–	**PRE**vious **POS**ition. No. of previous step.	–
35	STEPPOS	system	–	–	Current **STEP POS**ition. No. of current step.	–
64.156	NXTPOS	system	–	–	**N**e**XT** step **POS**ition. No. of next step.	–
64.158 : 64.161	JPOSx	system	–	–	**J**ump **POS**ition **x**. For x=1..4. Position of the step to which a jump will take place if condition x is fulfilled.	–
64.136	BLANKINF	system	–	–	**BLANK INF**ormation text and value. Erases information text and value. (INFTXT and INFVAL).	–
64.135	ALLACT	system	–	–	**ALL ACT**ivities are finished.	–
64.131 : 64.134	ACTSTAx	system	–	–	**ACT**ivity **STA**tus **x**. For x=1..4. Activity x is finished.	–

Op. step info 2



Terminal Description, Op. step info 2

Terminal No	Terminal Name	Value entered by	Default value	PC connection data type	Description	Remarks
64.175 : 64.182	CONDTXTx	system	–	–	Transition **COND**ition T**e**XT **x**. For x=1..8. Text describing transition condition x for current step.	–
64.183 : 64.186	JCTXTx	system	–	–	**J**ump **C**ondition T**e**XT **x**. For x=1..4. Text describing jump condition x for current step.	–
64.123 : 64.130	CONDSTAx	system	–	–	Transition **COND**ition **STA**tus **x**. For x=1..8. Transition condition x is fulfilled.	–
64.139 : 64.142	JCSTAx	system	–	–	**J**ump **C**ondition **STA**tus **x**. For x=1..4. Jump condition x is fulfilled.	–

3.5 Operator Functions

The Operator functions are divided in principle into three parts:

- Presentation
- Dialog
- Event handling

3.5.1 Presentation

Display elements which can be used for different display types are available for use in the functional unit Sequence.

The display elements show the status and the controls of the process with different degrees of detail and is intended for the following displays:

- Object display
- Group display
- Object closeup display
- Process display.

Examples of different display elements which could be used in these displays are given in the following section. A complete list of the prepared elements available is provided in the manual *AdvaCommand User Interface Reference Manual.*

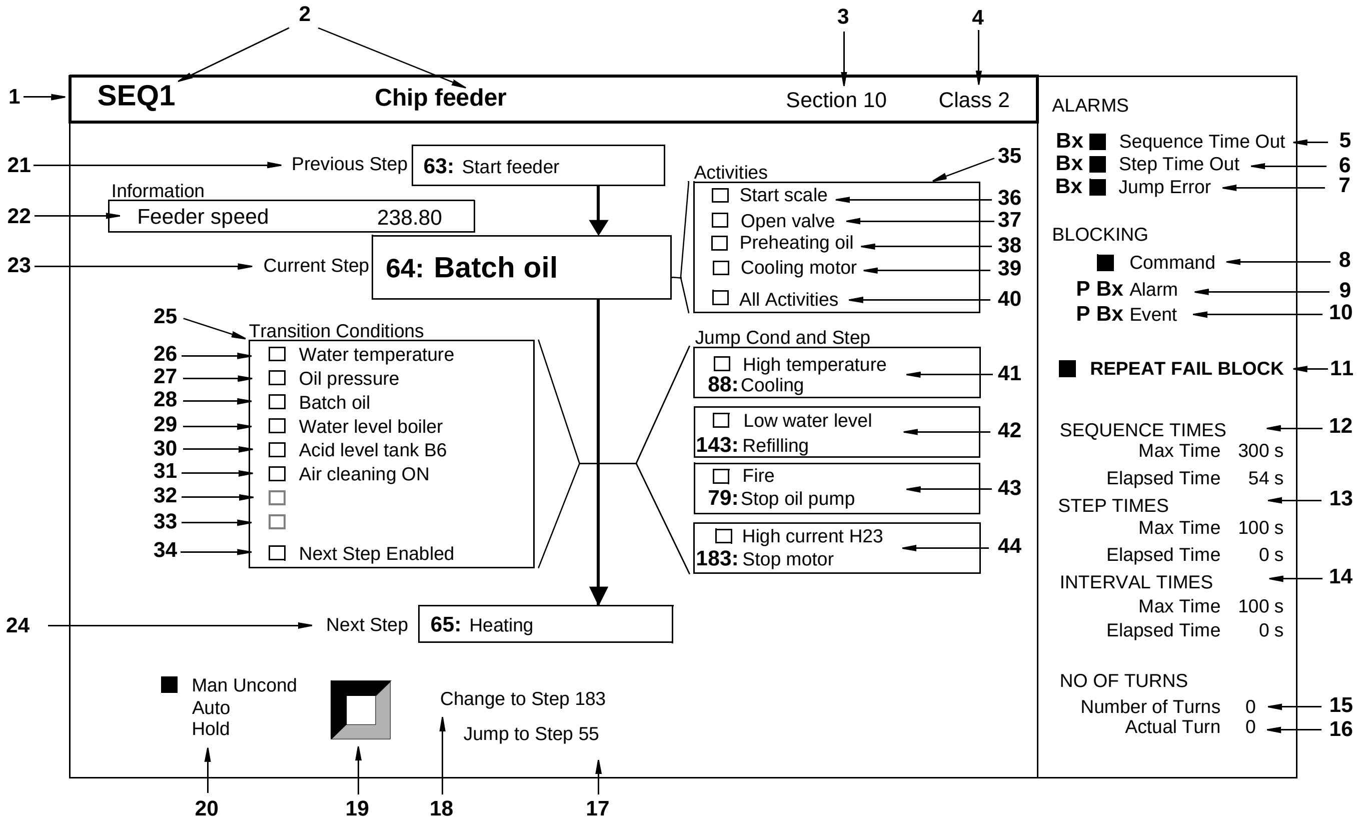
Detailed information about alternative ways of presenting certain status and data within the displays such as color, flashing, blocked etc. is given in the reference manual.

Sequence control often requires display of specific information about the current step, e.g. whether the stepping conditions are satisfied, whether the different activities of the step have been executed, and the identities of the preceding and succeeding steps.

Thanks to this kind of information, the operator is able to intervene quickly in case obstacles arise. This information is provided auto-matically by the object display, provided the required parameters have been filled into the Operator functions part of the structural element STEP.

This type of information may also be put into application-specific process diagram and tabular-status displays with great flexibility. In this latter case, the information is managed by application programs in the Advant Controllers 400 Series. In the the Operator Stations, static display creation and display elements for calculated DIs, DOs or DATs are used.

Object display: SEQOB01



No	Description	Remarks
1	SelectFrame	
2	Header	Object name and description
3.1	Section Text	
3.2	Section Value	
4.1	Class Text	
4.2	Class Value	
5	Sequence Time Out	
5.1	Alarm Blocked	
5.2	Alarm Indication	Unacknowledged alarm
5.3	Sequence Time Out Text	
6	Step Time Out	
6.1	Alarm Blocked	
6.2	Alarm Indication	Unacknowledged alarm
6.3	Step Time Out Text	
7	Jump Error	

No	Description	Remarks
7.1	Alarm Blocked	
7.2	Alarm Indication	Unacknowledged alarm
7.3	Jump Error Text	
8	Command Blocking	
8.1	Command Blocked	
8.2	Command Text	
9	Alarm Blocking	
9.1	Printout Blocked	
9.2	Alarm Blocked	
9.3	Alarm Text	
10	Event Blocking	
10.1	Printout Blocked	
10.2	Event Blocked	
10.3	Event Text	
11.1	Repeat Fail Block Indication	

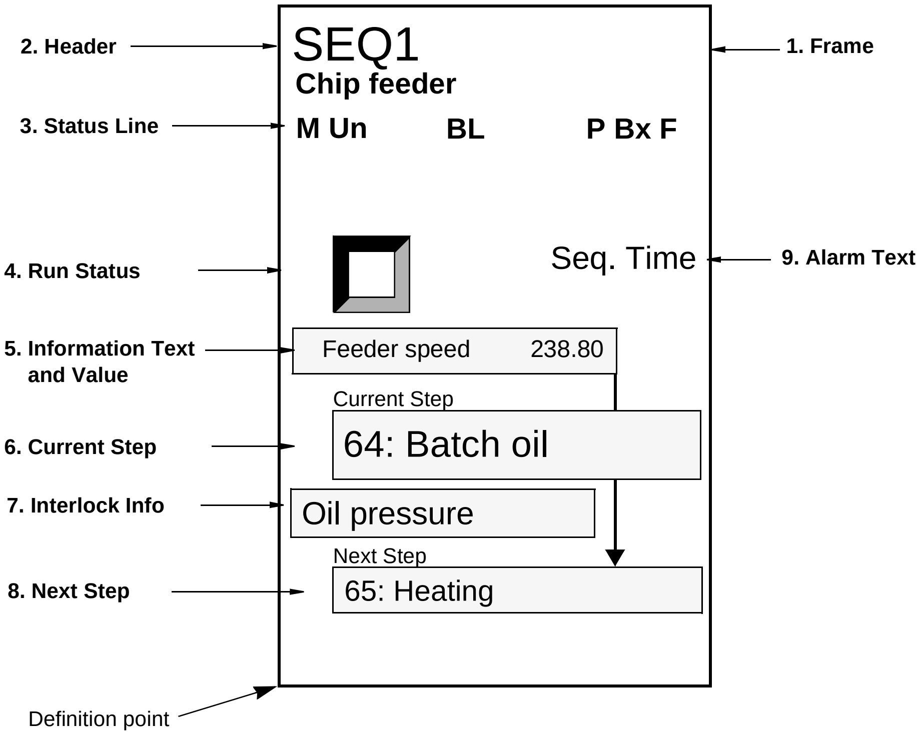
No	Description	Remarks
11.2	Repeat Fail Block Text	
12	SEQUENCE TIMES	
12.1	Max Sequence Time Text	
12.2	Max Sequence Time	
12.3	Elapsed Sequence Time Text	
12.4	Elapsed Sequence Time	
13	STEP TIMES	
13.1	Max Step Time Text	
13.2	Max Step Time	
13.3	Elapsed Step Time Text	
13.4	Elapsed Step Time	
14	INTERVAL TIMES	
14.1	Max Interval Time Text	
14.2	Max Interval Time	
14.3	Elapsed Interval Time Text	
14.4	Elapsed Interval Time	
	NO OF TURNS	
15.1	Number of Turns Text	
15.2	Number of Turns	
16.1	Actual Turn Text	
16.2	Actual Turn	
17.1	Jump to Step Text	
17.2	Jump to Step Value	
18.1	Change to Step Text	
18.2	Change to Step Value	
19	Run Status Indication	
20.1	Manual mode Indication	
20.2	Auto mode	
20.3	Hold mode	
21.1	Previous Step Text	
21.2	Previous Step Box	
21.3	Previous Step Number and Name	
22.1	Information Text	
22.2	Information Box	
22.3	Information Text	
22.4	Information Value	
23.1	Current Step Text	
23.2	Current Step Box	
23.3	Current Step Number and Name	
24.1	Next Step Text	
24.2	Next Step Box	
24.3	Next Step Number and Name	
25.1	Transition Condition Text	
25.2	Transition Condition Box	

No	Description	Remarks
26.1	Transition Condition 1 Ind.	
26.2	Transition Condition 1 Text	
27.1	Transition Condition 2 Ind.	
27.2	Transition Condition 2 Text	
28.1	Transition Condition 3 Ind.	
28.2	Transition Condition 3 Text	
29.1	Transition Condition 4 Ind.	
29.2	Transition Condition 4 Text	
30.1	Transition Condition 5 Ind.	
30.2	Transition Condition 5 Text	
31.1	Transition Condition 6 Ind.	
31.2	Transition Condition 6 Text	
32.1	Transition Condition 7 Ind.	
32.2	Transition Condition 7 Text Text	
33.1	Transition Condition 8 Ind.	
33.2	Transition Condition 8	
34.1	Next Step Enabled Indication	
34.2	Next Step Enabled	
35.1	Activities Text	
35.2	Activities Box	
36.1	Activity 1 Indication	
36.2	Activity 1 Text	
37.1	Activity 2 Indication	
37.2	Activity 2 Text	
38.1	Activity 3 Indication	
38.2	Activity 3 Text	
39.1	Activity 4 Indication	
39.2	Activity 4 Text	
40.1	All Activities Indication	
40.2	All Activities Text	
41	Jump Cond and Step Text	
41.1	Jump Condition 1 Box	
41.2	Jump Condition 1 Indication	
41.3	Jump Condition 1 Text	
41.4	Jump Step 1 Number and Name	
42.1	Jump Condition 2 Box	
42.2	Jump Condition 2 Indication	
42.3	Jump Condition 2 Text	
42.4	Jump Step 2 Number and Name	
43.1	Jump Condition 3 Box	
43.2	Jump Condition 3 Indication	
43.3	Jump Condition 3 Text	
43.4	Jump Step 3 Number and Name	
44.1	Jump Condition 4 Box	

No	Description	Remarks
44.2	Jump Condition 4 Indication	
44.3	Jump Condition 4Text	

No	Description	Remarks
44.4	Jump Step 4 Number and Name	

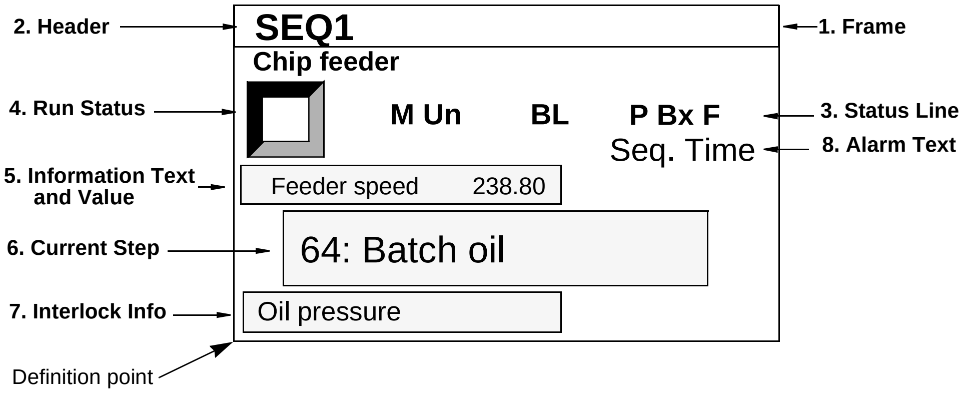
Display Element for Group display: SEQGR01



No	Description	Remarks
1	SelectFrame	
2	Header	Object name and description
3	Status Line	
3.1	Mode	
3.2	Printout Blocked	
3.3	Alarm Blocked	
3.4	Fault Indication	
3.5	Command Blocked	
4	Run Status Indication	
5	Information Box	

No	Description	Remarks
	Information Text	
	Information Value	
6.1	Current Step Text	
	Current Step Box	
	Current Step Number and Name	
7	Interlock Info Text	
8	Next Step Text	
	Next Step Box	
	Next Step Number and Name	
9	Alarm Text	

Display Element for Object closeup display: SEQWI01

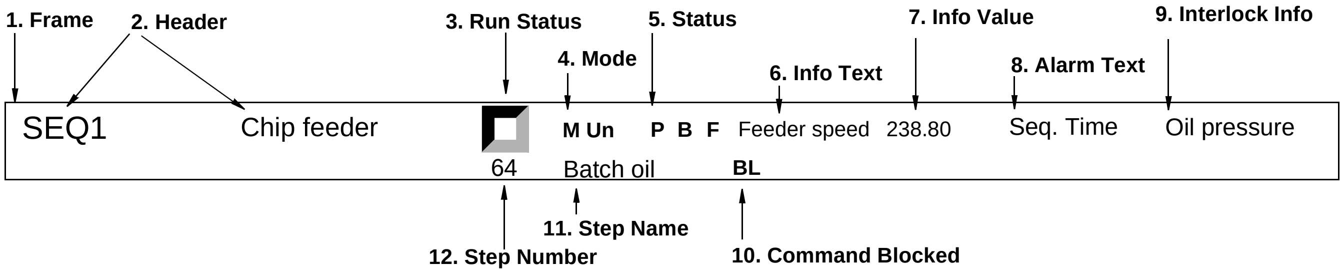


No	Description	Remarks
1	SelectFrame	
2	Header	Object name and description
3	Status Line	
3.1	Mode	
3.2	Printout Blocked	
3..3	Alarm Blocked	
3.4	Fault Indication	
3.5	Command Blocked	

No	Description	Remarks
4	Run Status Indication	
5	Information Box	
	Information Text	
	Information Value	
6	Current Step Box	
	Current Step Number and Name	
7	Interlock Info Text	
8	Alarm Text	

Display Element for Process display:

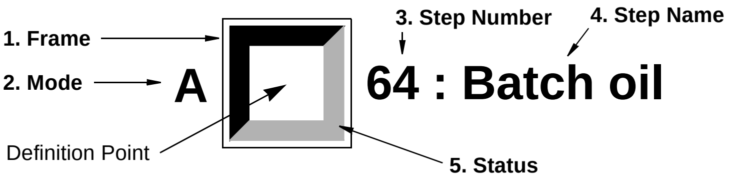
SEQSG01



No	Description	Remarks
1	SelectFrame	
2	Header	Object name and description
3	Run Status Indication	
4	Mode	
5	Printout Blocked	
	Alarm Blocked	
	Fault Indication	

No	Description	Remarks
6	Information Text	
7	Information Value	
8	Alarm Text	
9	Interlock Info Text	
10	Command Blocked	
11	Current Step Name	
12	Current Step Number	

SEQPD01



No	Description	Remarks
1	SelectFrame	
2	Mode	
3	Current Step Name	

No	Description	Remarks
4	Current Step Number	
5	Run Status Indication	

3.5.2 Dialog

The display screen is supplemented with a keyboard for operator communication with the functional unit/sequence.

The dialog is built up using both fixed and dynamic keys. The functions available with the dynamic keys are presented in a pattern of rectangles on the lower part of the display screen when the object is selected. The functional unit Sequence has a MIN dialog (Production level) and a MAX dialog (Maintenance level). The MIN dialog consists of an output dialog but no subdialogs. The MAX dialog consists of an output dialog with 2 subdialog.

MIN dialog (No authority key)

Keyboard keys

◇	**Start**	Start the Sequence
▽	**Stop**	Stop the Sequence
Man	**Man**	Set to manual mode
Auto	**Auto**	Set to automatic mode
✓	**Acknowledge**	Acknowledge alarm
Object	**Object Display**	Select Object Display
Dialog	**Previous Level**	Select Top Level dynamic keys

Dynamic keys, Top Level

D1	D2 ⊟ Hold	D3 Step	D4 Reset	D5
D6	D7	D8	D9	D10 *Keyboard…*

Hold	Set to hold mode
Step	Steps the Sequence one step
Reset	Reset the Sequence to initial values
Keyboard…	Select Dynamic keys: Keyboard

Dynamic keys, Keyboard

D1 ■ Start	D2 Man	D3 Auto	D4	D5
D6 □ Stop	D7 Acknowledge	D8 Object Display	D9	D10

Start	Start the Sequence
Man	Start the Sequence
Auto	Set to automatic mode
Stop	Stop the Sequence
Acknowledge	Acknowledge alarm
Object Display	Select Object Display

MAX dialog (Authority key)

Keyboard keys

◇	**Start**	Start the Sequence
▽	**Stop**	Stop the Sequence
Man	**Man**	Set manual mode
Auto	**Auto**	Set automatic mode
✓	**Acknowledge**	Acknowledge alarm
Object	**Object Display**	Select Object Display
Dialog	**Previous Level**	Select Top Level dynamic keys
Dialog	**Next Block**	Page between the dynamic key sets (Keyboard, Block and Parameters)

Dynamic keys, Top Level

D1	D2 ⊟ Hold	D3 Step	D4 Reset	D5 Jump
D6	D7	D8 *Block…*	D9 *Parameters…*	D10 *Keyboard…*

Hold	Set to hold mode
Step	Steps the Sequence one step
Reset	Reset the Sequence to initial values
Jump	Jump to step indicated by parameter
Block…	Select Dynamic keys: Block
Parameters…	Select Dynamic keys: Parameters
Keyboard…	Select Dynamic keys: Keyboard

Dynamic keys, Keyboard...

D1 ■ Start	D2 Man	D3 Auto	D4	D5
D6 □ Stop	D7 Acknowledge	D8 Object Display	D9	D10

Start	Start the Sequence
Man	Set to manual mode
Auto	Set to automatic mode
Stop	Stop the Sequence
Acknowledge	Acknowledge alarm
Object Display	Select Object Display

Dynamic keys, Block...

D1 Block Command	D2 Block Alarm	D3 Block Printout	D4 Block I/P Event	D5 Block I/P Printout
D6 Deblock Command	D7 Deblock Alarm	D8 Debloc Printout	D9 Deblock I/P Event	D10 Deblock I/P Printout

Block Command	Block command
Block Alarm	Block alarm indication
Block Printout	Block event printout
Block I/P Event	Block I/P event indication
Block I/P Printout	Block I/P event printout
Deblock Command	Deblock command
Deblock Alarm	Deblock alarm indication
Deblock Printout	Deblock event printout
Deblock I/P Event	Deblock I/P event indication
Deblock I/P Printout	Deblock I/P event printout

Dynamic keys, Parameters...

D1 Conditional	D2 Jump Pos	D3 No of Turns	D4 Interval Time	D5
D6 Unconditional	D7	D8	D9	D10

Conditional	Condition for manual step on
Jump Pos	Check/change the step number that the sequene jumps to if JUMP is pushed
No of Turns	Number of turns for the sequence to run
Interval Time	Control/change the interval between two sequences
Unconditional	Condition for manual step off

3.5.3 Event Handling

This section contains a description of all standard alarms and events in the functional unit Sequence. The description is given under the following headings.

- General
- Event description with standard texts
- Event and alarm blocking.

Event handling as a common property of the Advant OCS system is described in detail in the AdvaCommand documentation. Reference is made for example to the manual *AdvaCommand Basic Functions User's Guide* when there are special requirements of user defined event handling incl. texts.

The event texts for all functional units are also collected in the manual *AdvaCommand User Interface Reference Manual*, giving the user a convenient quick reference library for the application work.

General

The event handling is controlled by some configuration parameters, EVENT-TREAT pointers, in the data base element located in Advant Controller 400 Series and by utilizing a standard or a user defined Event Treat data base element in Advant Station 500 Series Operator Station. The principle is illustrated in the figure below.

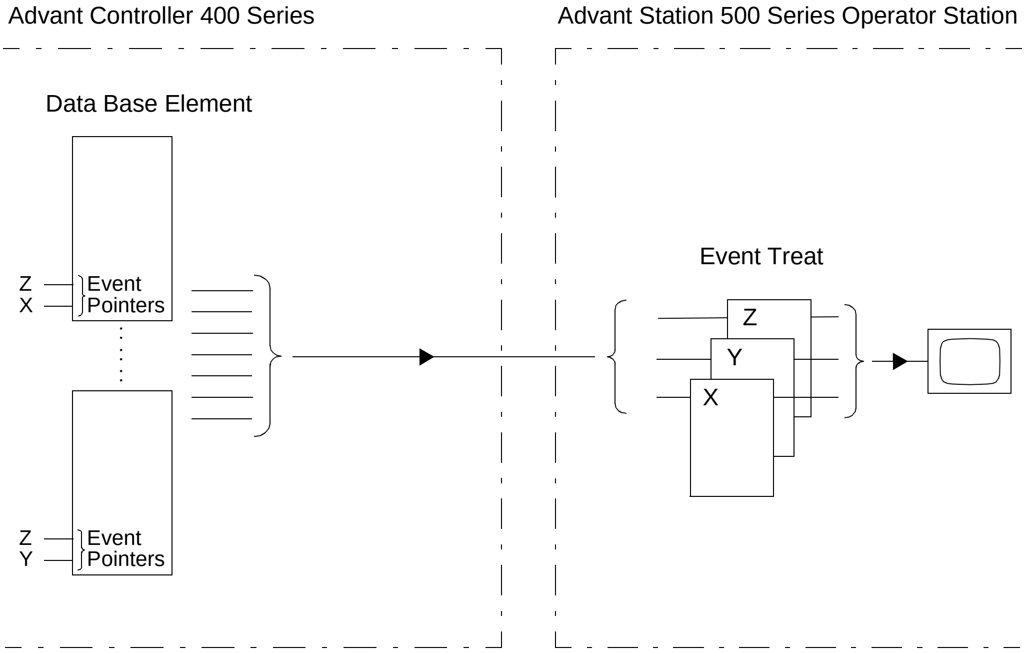

Advant Controller 400 Series
Advant Station 500 Series Operator Station
Data Base Element
Z
X
Event Pointers
Z
Y
Event Pointers
Event Treat
Z
Y
X

Figure 3-8. The principle of event handling

The Event Treat defines the event handling in the operator station such as audible alarms, text combinations in event and alarm list, colors, different handling for 0 -->1 and 1 -->0 transitions.

How different events from SEQ is treated is given in Table 3-1 below. This also specifies which event handling is obtained as default and which alternative standard functions are available.

Event description with standard texts

Table 3-1. Event Texts for SEQ

Events	Event Treat pointer	Property text		Event text	
		Standard text	Line No in Event Treat	Standard text	Line No in Event Treat
Change the step	POS_TREAT	Step change	1	Value change	1
Indication Active On	IND_TREAT	Active	1	On	1
Indication Active Off	IND_TREAT	Active	1	Off	2
Indication Last Step On	IND_TREAT	Last Step	2	On	3
Indication Last Step Off	IND_TREAT	Last Step	2	Off	4
Indication Auto control mode On	IND_TREAT	Auto Ind	3	On	5
Indication Manual control mode On	IND_TREAT	Man Ind	4	On	5
Indication step Hold On	IND_TREAT	Hold Ind	5	On	5
Indication Unconditional control mode On	IND_TREAT	Uncond Ind	6	On	5
Indication Next step On	IND_TREAT	Next	7	On	5
Position error On	FAULT_TR	Jump error	1	Alarm	1
Position error Off	FAULT_TR	Jump error	1	Normal	2
Sequence error On	FAULT_TR	SeqAlarm	3	Alarm	1
Sequence error Off	FAULT_TR	SeqAlarm	3	Normal	2
Step error On	FAULT_TR	StepAlarm	4	Alarm	1
Step error Off	FAULT_TR	StepAlarm	4	Normal	2

Event and Alarm Blocking

The event and alarm blocking is discussed in a general way in the manual *Functional Units Part 1 Common Properties.*

With special commands in the dialog for the functional unit Sequence, the operator can block the listed event- and alarm handling.

Table 3-2. Event and Alarm Blocking

Dialog Key	Event- and Alarm handling blocked for signals:	Remark
BLK AL	SEQAL, STEPAL, POSF	Block alarm
BLK PRT	SEQAL, STEPAL, POSF	Block print (Blocks printout on printer only)
BLK I/P EV	POS (STEP), RUN, END, AUTOM, MANM, HOLDM, UNCONDM, NEXT	Block I/P events
BLK I/P PRT	POS (STEP), RUN, END, AUTOM, MANM, HOLDM, UNCONDM, NEXT	Block print (Blocks printout on printer only)

3.5.4 Group Alarm

It is possible to connect the functional unit Sequence to the group alarm function. This function makes it possible to gather alarms and events from related process signals and objects into Alarm Groups. These groups are then treated as objects.

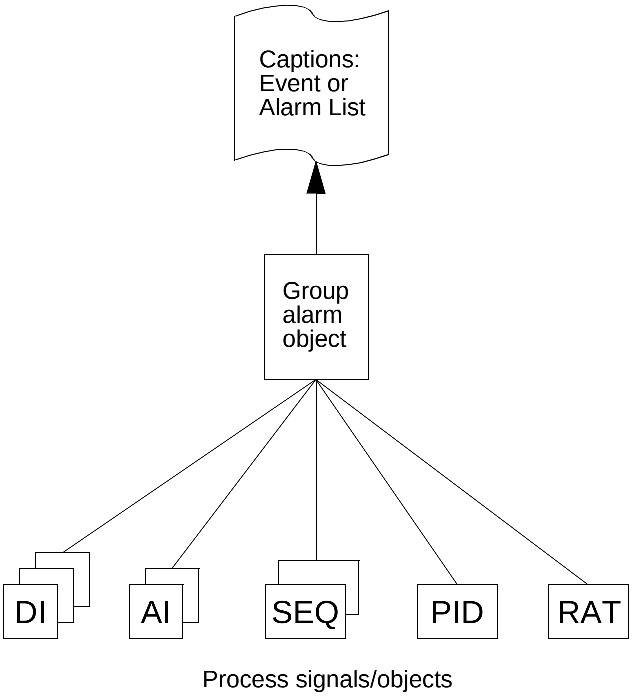


Figure 3-9. Group Alarm

The group alarm function consists of four parts as follows:

- Grouping of process alarms and events with a view to creating pilot alarms and events.
- Blocking of groups
- Automatic acknowledgement
- External alarm outputs

For a more detailed description of the group alarm function, reference is made to the manual *AdvaCommand Basic Functions User's Guide*.

3.6 Application Examples

The PC program is the combination of PC elements and structure elements fulfilling the automation and controlling requirements of the process specified by the application engineer.

This document is not intended to give a wide spectrum of application examples. This chapter indicates connection possibilities and requirements of a sequence. Please also refer to Section 3.2.10 Operator Functions, which give a common information of connection possibilities to data base and the operator station dialog and display facilities.

Sequence control is a basic function of the Advant OCS system. Therefore a lot of detailed application oriented instructions and structuring rules for sequences are included in the separate manual *AMPL Application Building Reference Manual*. This book is a basic manual for the ABB Master Programming Language.

3.6.1 Example 1

This example shows the minimum requirement of connections to the database element when using the Operator functions.

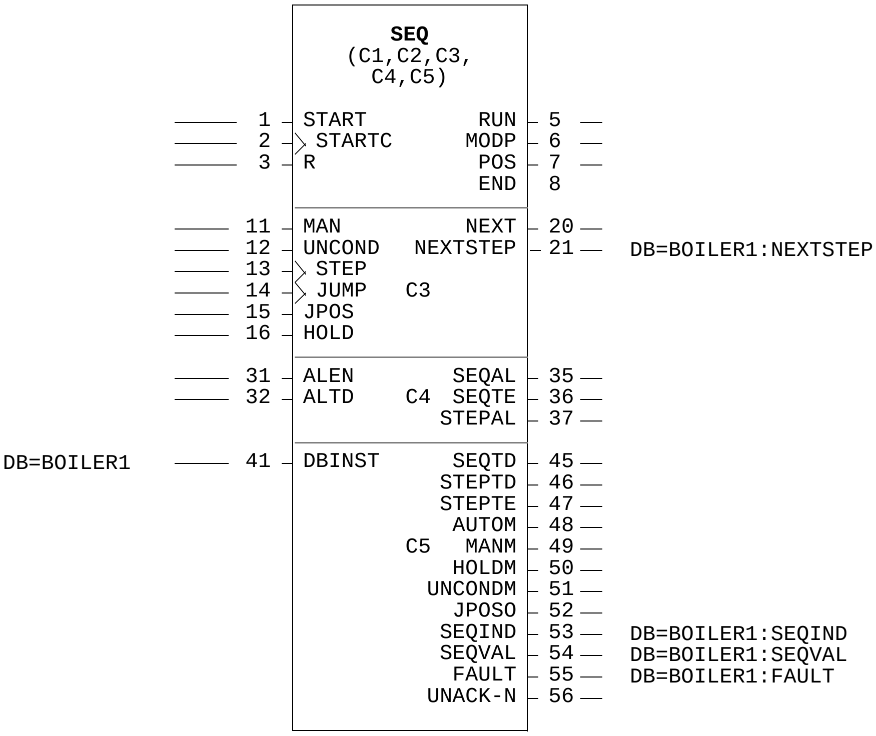


Figure 3-10. PC element SEQ

The operator interaction in this example is limited to a minimum. He can not give any orders to the sequence. The only interaction is the blocking feature, see details in the Section 3.5.3 Event Handling, covering the operator dialog for Event and alarm blocking. However all the indications of the sequence will be performed via the connections. The orders and the mode selections have to be fulfilled by PC elements in the PC program.

3.6.2 Example 2

This second example shows a normal connection requirement. All the operator orders have to be connected individually from the PC element to the database element.

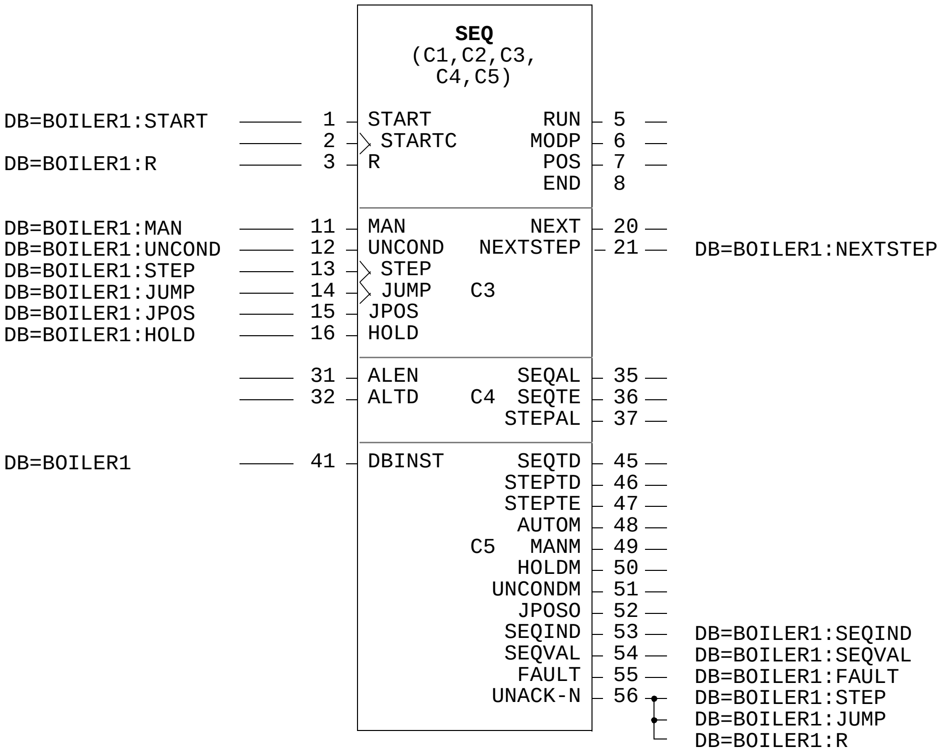


Figure 3-11. PC element SEQ

The following comments are useful:

- ALEN and ALTD have to be set during PC programming using constants, operational parameters or outputs of other PC elements.
- UNACK_N is connected with the data base terminal of STEP, JUMP and R, to reset the operator order after receiving the information.

3.6.3 Example 3

The third example shows a configuration, where operator orders and conditional orders are mixed. The conditional orders are accomplished by the use of standard PC elements.

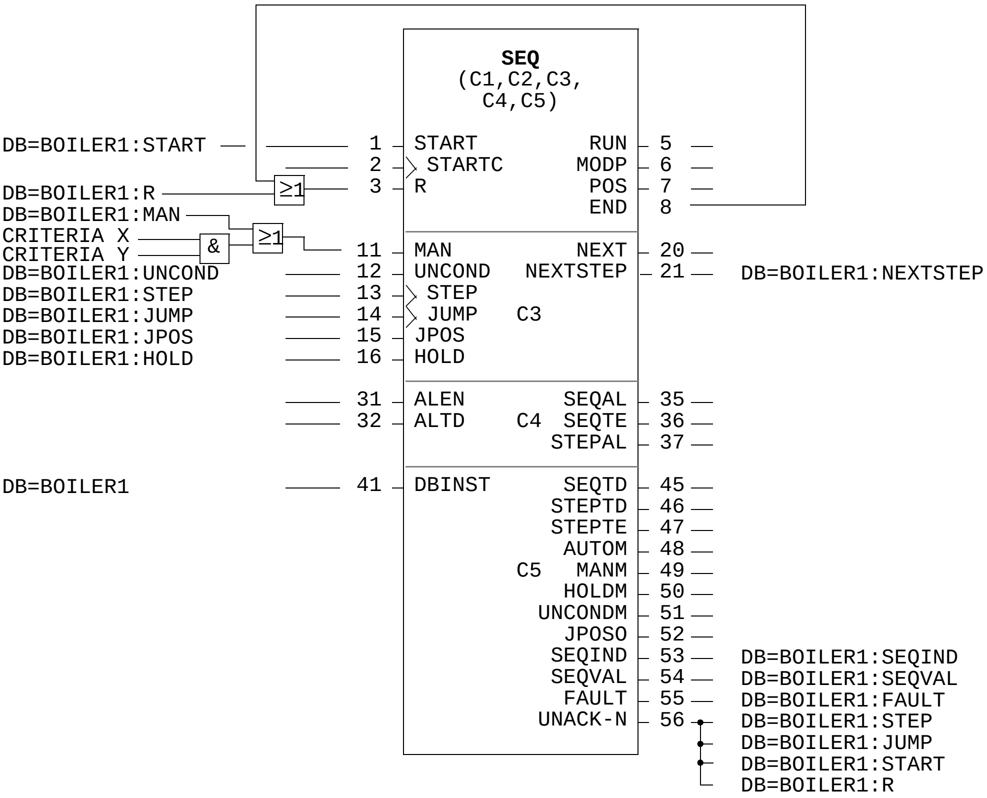


Figure 3-12. PC element SEQ

The purpose of the third example is to show

- The use of conditional orders
- That SEQ database element terminals can be connected to other PC elements to mix operator and conditional orders
- That the UNACK_N output can and must be connected to a variety of database terminals depending on the application requirements
- That operator orders are not sent in a block terminal similar to SEQIND, because the individual connections give the application engineer the freedom of using conditional orders wherever necessary for the application

A brief description of the function:

The sequence will run once and be ready for the next start order from the operator. There is no need of a reset order from the operator (END connected to R). However, the operator must always have the possibility to reset the sequence whenever necessary. (The use of the OR-gate.)

The application can require a man mode selection as a security measure. (The use of the AND-gate with criteria X and Y.)

3.6.4 Example 4

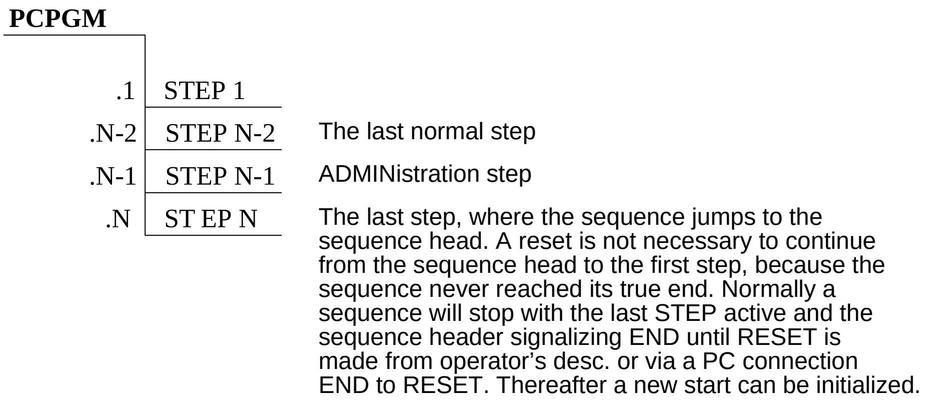
The last example shows an application where the number of turns for a sequence should be counted and where the interval time delay between turns have to be supervised.

This example shows a sequence application where the program functions associated to the properties TURNS and INTV_TD are placed inside the sequence structure.

The indicated status will be RUN, when the sequence has completed a turn and is waiting to start for the next. Other solutions are possible. Sometimes, with respect to simplicity and changeability, there is an advantage with the supervision program placed outside the sequence structure. The disadvantage is, that the sequence status is shown as OFF in the displays, when the sequence is in the “semiactive” status, counting the interval time elapsed.

The PC structure calculating the TURNS and INTV_TD is in this case placed inside the structure of the next but the last STEP. This could be called the ADMINistration step.

The program has the following structure:



The last normal step

ADMINistration step

The last step, where the sequence jumps to the sequence head. A reset is not necessary to continue from the sequence head to the first step, because the sequence never reached its true end. Normally a sequence will stop with the last STEP active and the sequence header signalizing END until RESET is made from operator’s desc. or via a PC connection END to RESET. Thereafter a new start can be initialized.

Figure 3-13. Example 4, Program structure

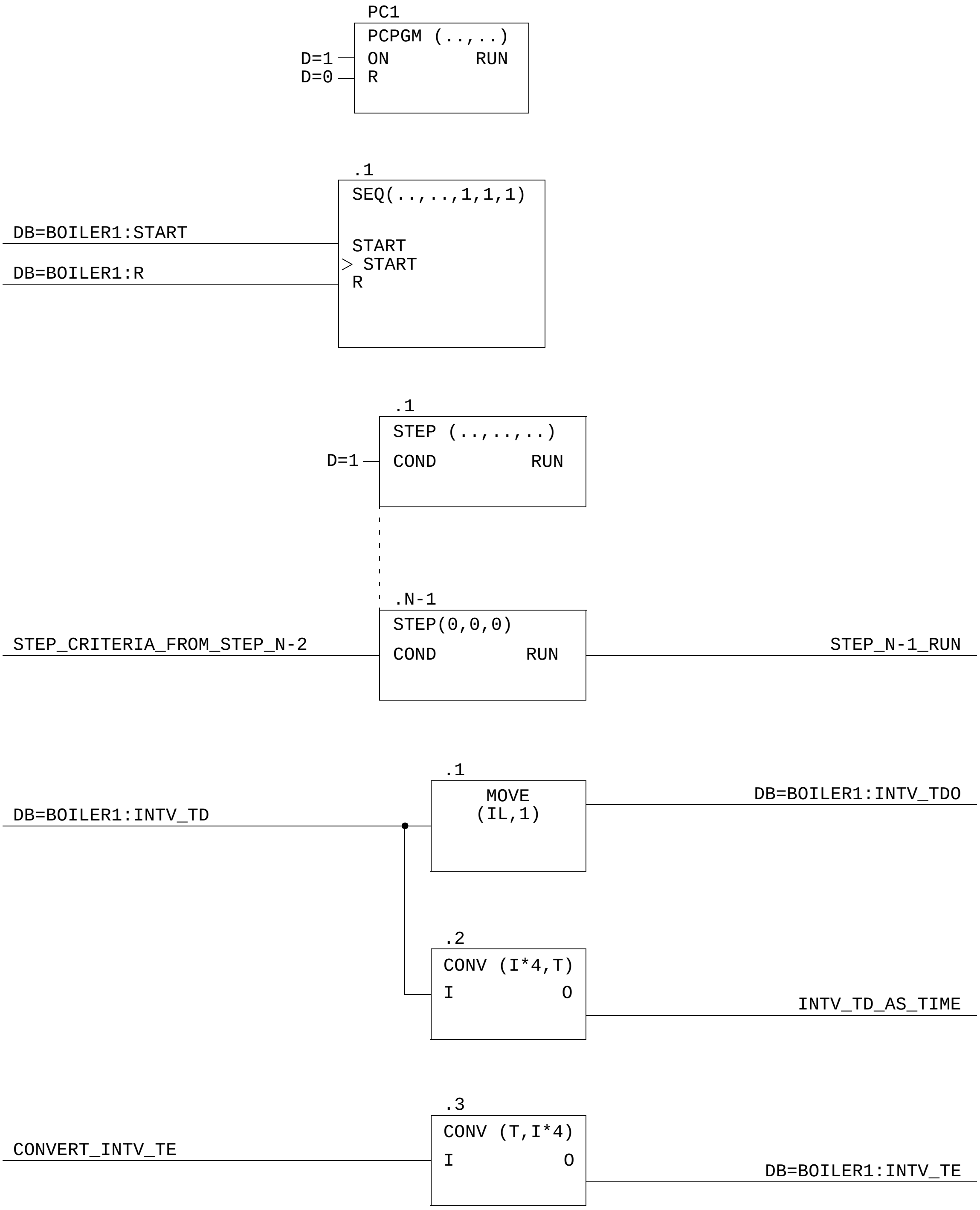


Figure 3-14. PC1.1.N-1 (1 of 2)

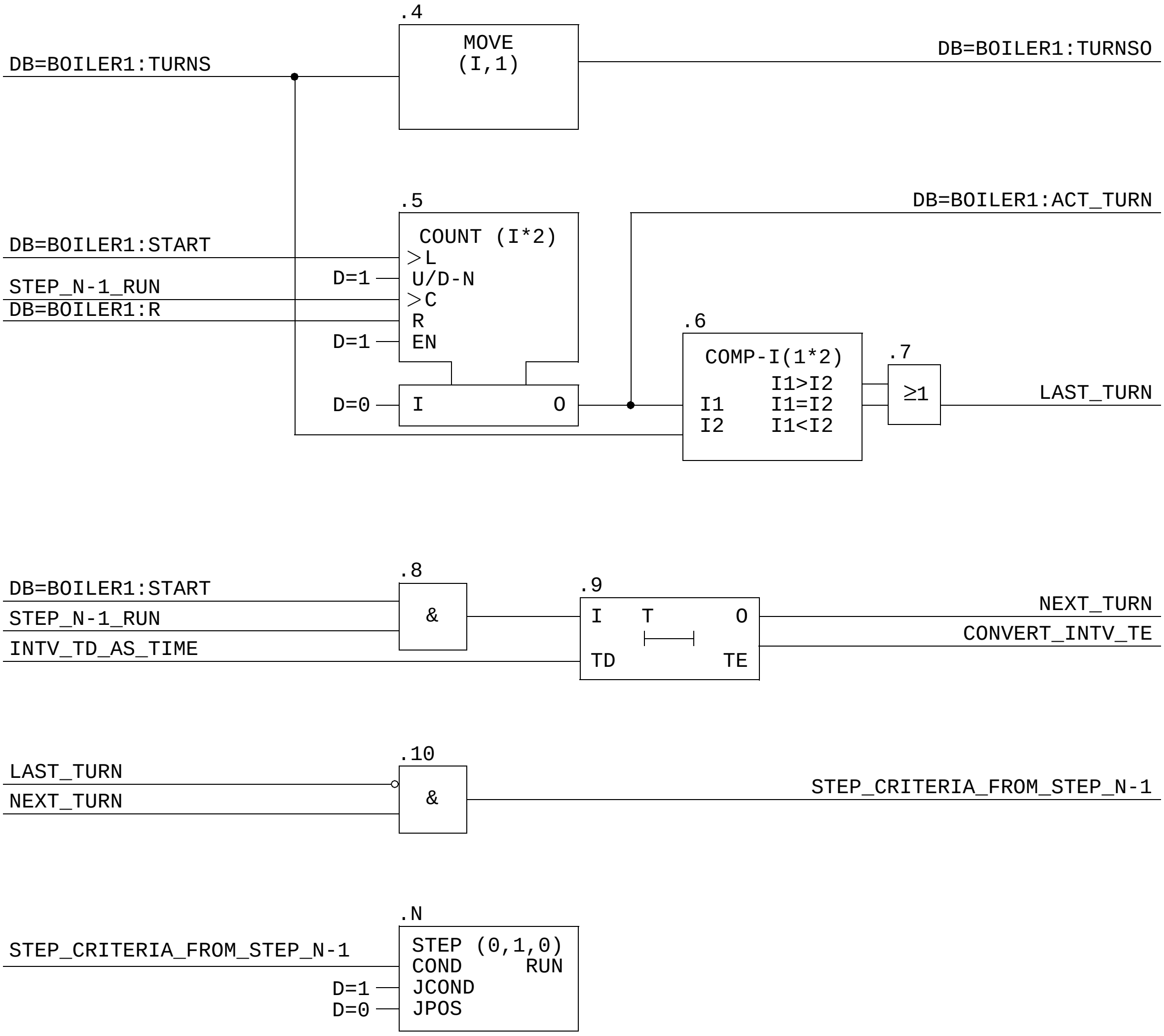


Figure 3-15. PC1.1.N-1 (2 of 2)

3.6.5 Example 5

This example shows how the object display of the sequence is kept updated by the structural element STEP. The Operator functions section of STEP contains text strings that are shown on the object display. When the sequence advances to the next step, the display changes contents.

Figure 3-16 shows a wagon that is being filled and weighed. The figure also shows the signal exchange with the process.

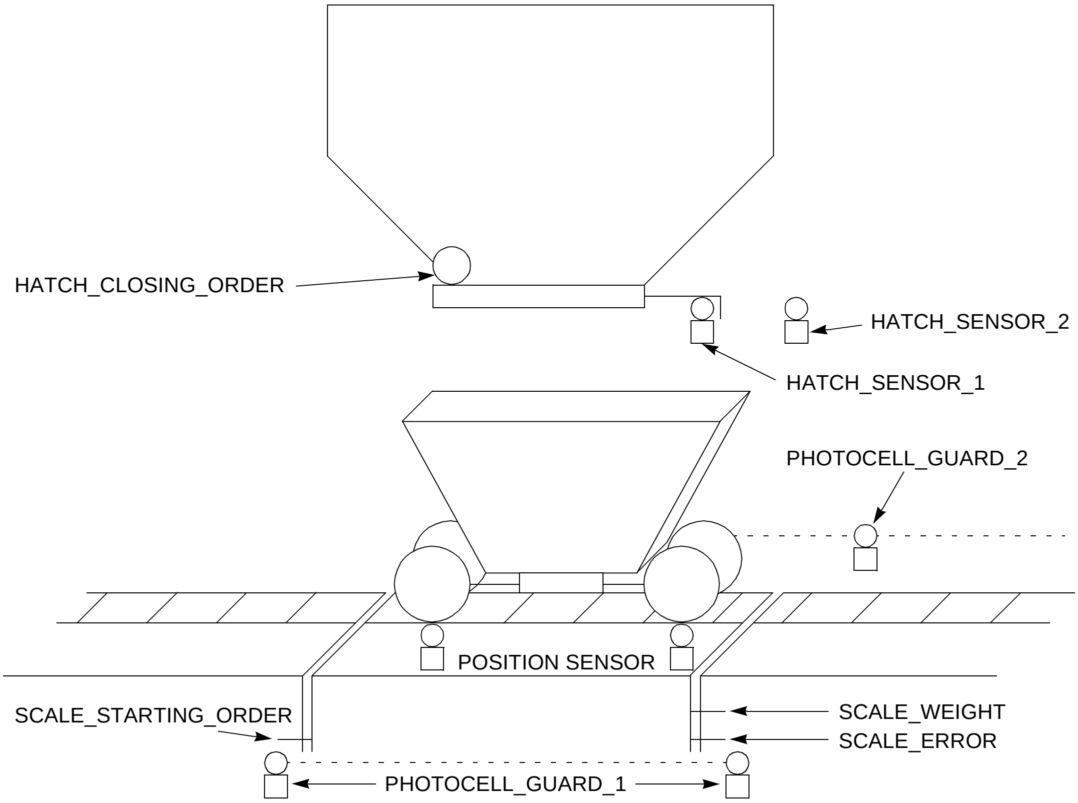


Figure 3-16. Wagon being filled and weighed

Figure 3-17 shows a functional diagram of the sequence SEQ1, Loading. This sequence starts with the following steps:

1. Wagon forwards, brings the wagon up to the filling position
2. Filling, comprising the following activities:

   Initiation of weighing by sending a start order (SCALE_STARTING_ORDER) to the weighing equipment.

   Filling of the wagon by opening the hatch. The signal HATCH_SENSOR_2 shows whether the hatch is open.

   End of filling of wagon by closing the hatch (HATCH_CLOSING_ORDER)

3. Filling, requests advance to step 3, Wagon Out, when the following conditions are satisfied:

   The wagon has reached the weight 700 kg (SCALE_WEIGHT)

   The hatch has closed (HATCH_SENSOR_1)

The sequence SEQ1, Loading, terminates its normal operation and jumps from step 2 if any of the following conditions are met:

- An operator comes close to the wagon (PHOTOCELL_GUARD_1).
- The wagon does not assume its correct position (POSITION_SENSOR).
- The weighing equipment issues an error signal (SCALE_ERROR).
- The tracks of the wagon are blocked (PHOTOCELL_GUARD_2).

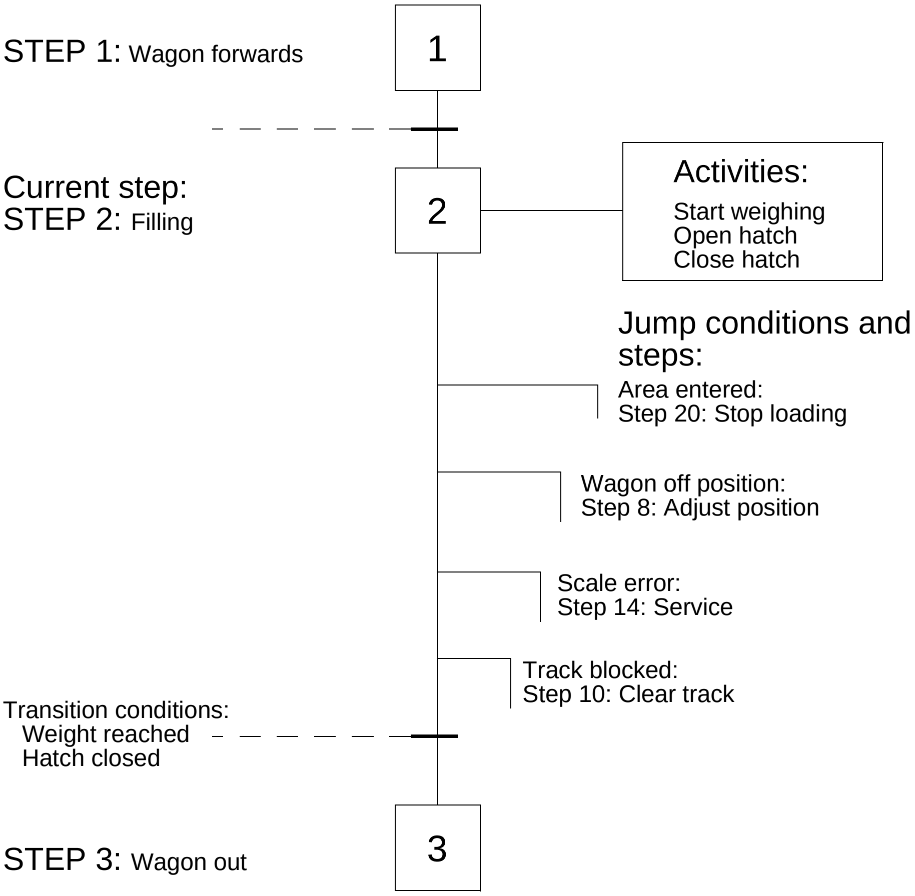


Figure 3-17. Functional diagram of the sequence “Loading”

Figure 3-18 shows the structural element STEP for step 2, Filling.

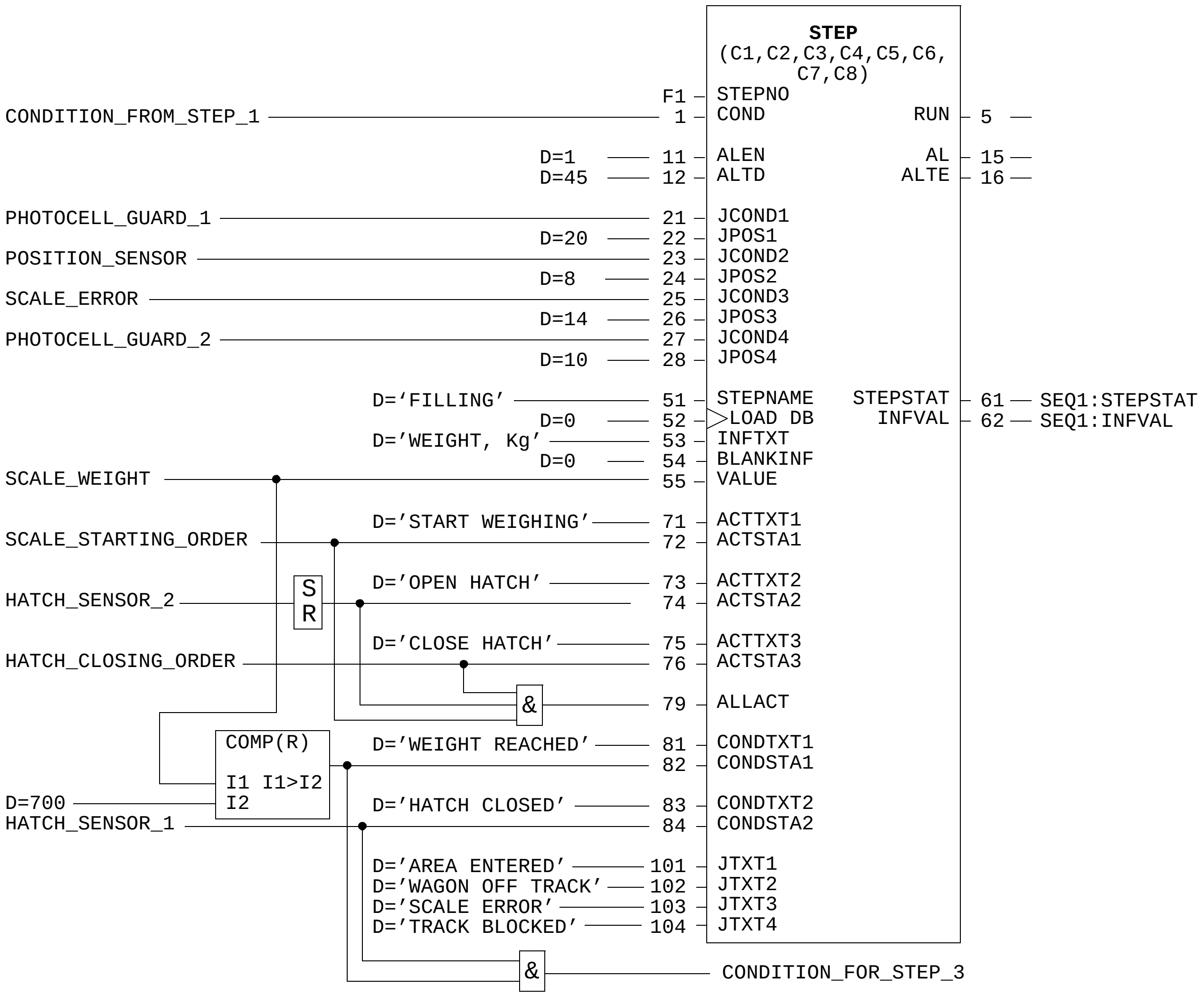


Figure 3-18. PC element STEP

Figure 3-19 shows the corresponding object display for the sequence Loading. The Operator functions section of STEP, terminals 51 and upwards, are shown on the object display.

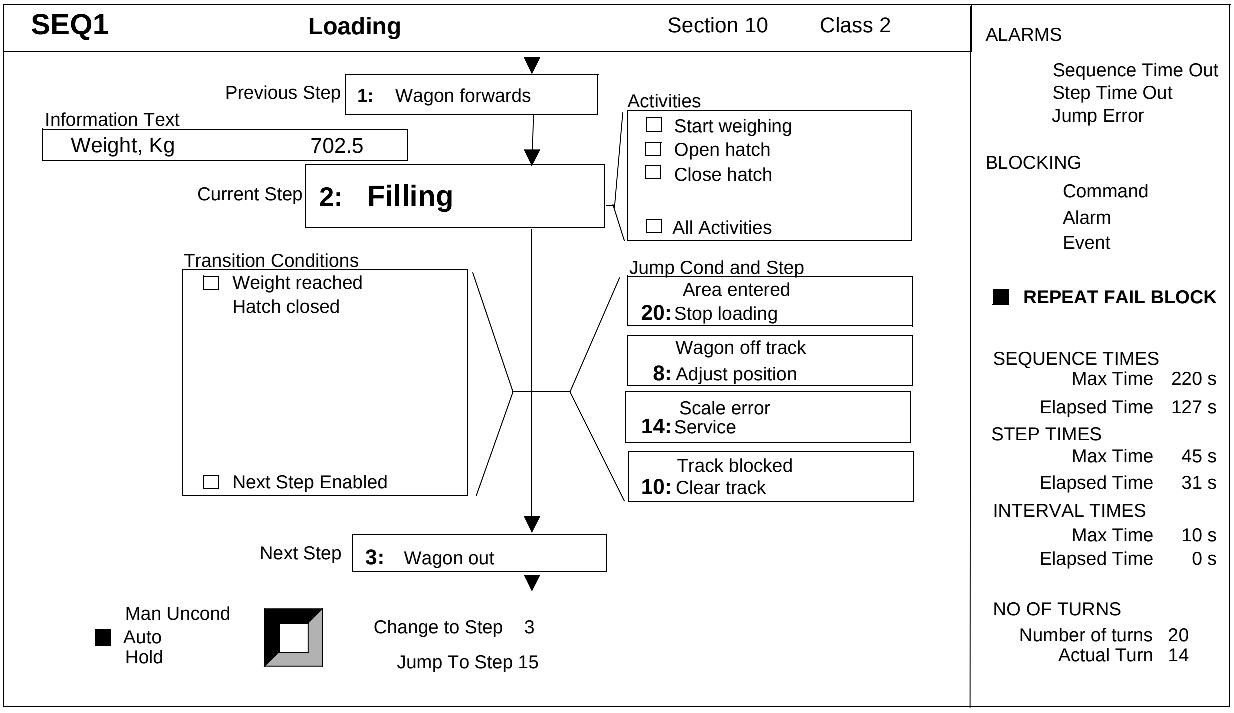


Figure 3-19. Object display of the sequence Loading when the current step is Filling

The object display of the sequence also shows information that goes beyond the Operator functions section of STEP as follows:

CHANGE TO STEP is only indicated in manual control mode. The number indicates the next step to be executed in manual mode if the operator requests stepping. This step represents the next step for normal execution or if the jump conditions are true, the jump destination with the highest priority.

JUMP TO STEP holds the destination step when a request for jumping is directed at the input JUMP on the header SEQ. JUMP TO STEP shows the contents of the input JPOS on the SEQ element.

See Section 3.5 Operator Functions, for a complete description of the object display for sequences.

Chapter 4 GROUP

The functional unit GROUP is used for group-start and group-stop of motors, valves and other binary objects coordinated as a group in the process. These objects are normally controlled by MOTCON or VALVECON with which GROUP is compatible.

The general functional unit GROUP is described in the following with respect to Operator functions and connection to PC program. The final chapter contains an example which shows an application program which supplements the functional unit to provide a complete group start.

4.1 Configuration

The functional unit GROUP consists of four parts:

1. A user defined application program which constitutes the design of the control function required. This program is built up of PC elements.
2. Two PC elements which are connection terminals for signals between data base element and PC.
3. A data base element which stores the signals described above and the values of design parameters.
4. Operator functions for display elements, dialogs and event treatment which constitute construction elements for use in displays and alarm signals.

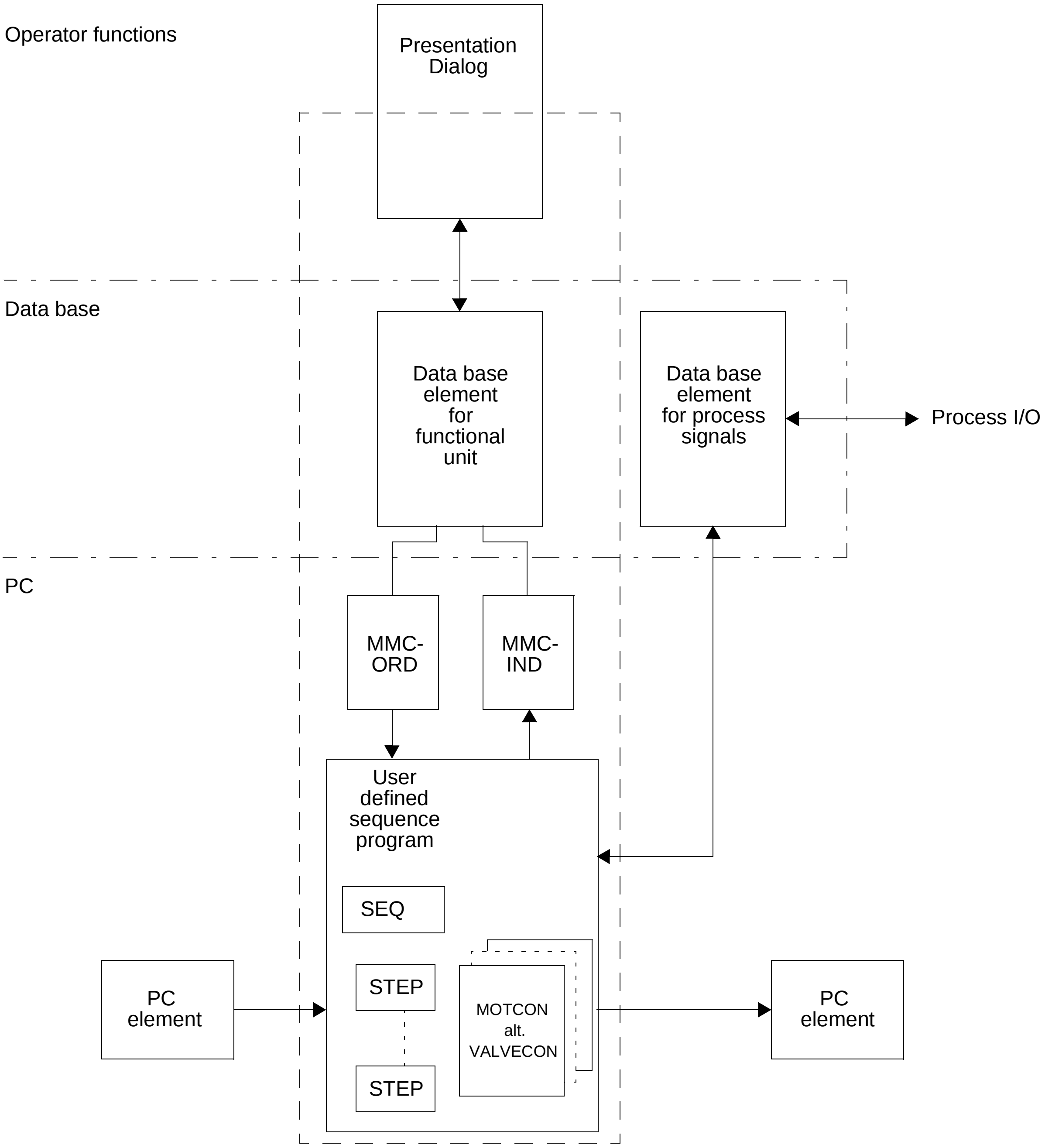


Figure 4-1. The structure of the functional unit GROUP

4.2 Function

The group start function consists of the functional unit GROUP and a PC function which can be freely chosen to provide the optimum function.

This means that it is not possible to prepare a general function description of group start but the proposed function for GROUP in a group start is described below.

The basic purpose of the PC function is to start or stop a group of objects in the process. The process is to be of the continuous type and it should not be started nor stopped too frequently, i.e. the process is controlled from a state of rest to a working status, something which is common in the process industry.

Examples of such application which are mainly of binary type are

- Starting/stopping motors
- Opening/closing valves or dampers
- Presetting or shut down of regulators controlling speed, flow, temperature, pressure etc.
- Positioning of machines to a working position or a parking position

In other words, the function is to simplify the work involved in preparing the process for production so that the operator only commands start or stop of the group instead of starting each object individually.

The group can also consist of subordinate groups on several levels depending on the degree of automation required.

The PC construction is ordinarily based on the elements SEQ and STEP.

The handling of the functional unit is similar to that of other binary functional units such as MOTCON and VALVECON, both for the operator and designer. This means that the use of certain points of control can appear difficult to understand. The intention is there that the higher level of control of, for example, a group of motors is not to differ from the control of an individual motor.

GROUP is intended to be handled from

- A display screen in the point of control **CENTRAL**
- Local control desk in the point of control **LOCAL**
- Master control function (e.g. group start) in the point of control **SEQUENCE**
- Push buttons immediately adjacent to the group in the point of control **TEST**, this reducing the scope of the supervision in the PC.

The control is completely disconnected in the point of control STANDBY.

The following control modes are assumed to be used in the LOCAL, CENTRAL and SEQUENCE points of control:

- **MAN**, when an operator starts the sequence directly
- **AUTO**, when an event in the process starts the sequence.

The following commands affect the sequence, irrespective of where they are given:

- START which starts the stepping in the sequence, either from the first step or from some other, depending on the current situation. Note that it can be a sequence which starts or stops the motors (the object).
- HOLD which stops the stepping in the sequence, in the step currently in execution.
- STOP can be used in either of two different ways.
  - To stop the sequence (and reset to the first step)
  - To stop the objects in the group (e.g. sequentially)
- FAST STOP which stops all of the objects in the group simultaneously.

The PC function can contain IC1 and 2, IB1 to 4 and IA interlockings which stop a sequence or object and are presented in displays with texts in the same way as when MOTCON and VALVECON are used. The operator blocks IB1 and IB3.

The display elements are intended to present the result from the PC calculation if for instance the group is Ready To Start or Ready To Stop. That means, most of the conditions for start of a motor (object) are assembled as a group of conditions common to the group of objects. It should not be possible to start the group if any of the objects cannot be started. This is characteristic of sequences of the group start type. Note that both the group status and individual motor status are shown in the Motor Group Display.

Further more a presentation is made in the display elements whether the objects are In Operation or Sequence In Progress and the Actual Step Number.

4.3 PC Elements

The general PC elements MMC-ORD for central operator’s commands and MMC-IND for presentations on display screen are included in the design of the functional unit GROUP.

These PC elements receive a specific GROUP- function significance for each connection.

Both elements are to be connected to the same data base element, MMCX.

4.3.1 PC Element MMC-IND

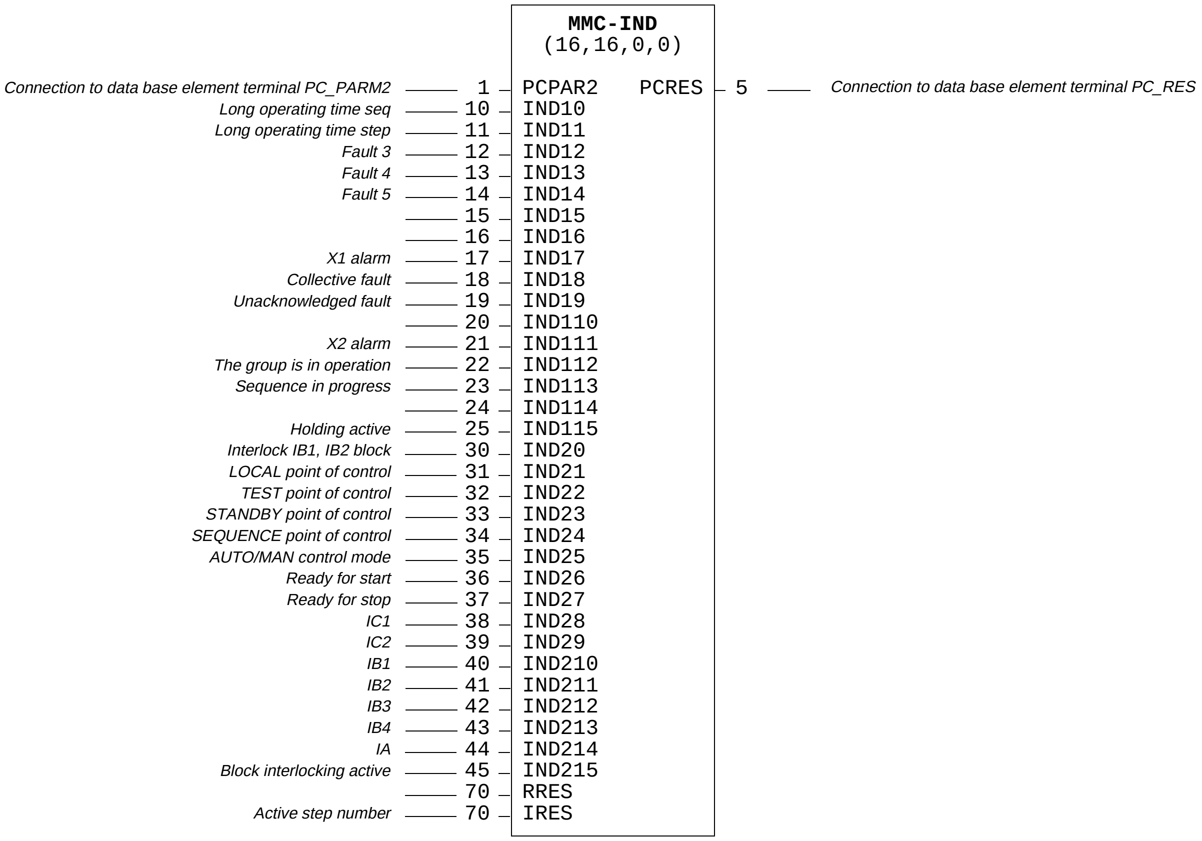
The figure below shows how the control signals are distributed to the different terminals on the PC element MMC-IND for the functional unit GROUP. This is followed by a brief supplementary explanation of the PC element. See also the preceding Section 4.2 Function.

Operator Station Indications MMC-IND

Summary

MMC-IND is used with the data base element MMCX, object handler, display elements of Advant Station and alarm and event handling. This enables the PC programmer to send indications and measured values from arbitrary PC programs as well as to generate alarms and events to the operator. The element can handle up to 32 indications, one real measured value and one integer measured value. The indications are divided into two groups (IND1 and IND2) with regard to alarm and event handling.

Picture Operator Station Indications element [(1)]


MMC-IND
(16,16,0,0)
Connection to data base element terminal PC_PARM2 1 PCPAR2
PCRES 5 Connection to data base element terminal PC_RES
Long operating time seq 10 IND10
Long operating time step 11 IND11
Fault 3 12 IND12
Fault 4 13 IND13
Fault 5 14 IND14
15 IND15
16 IND16
X1 alarm 17 IND17
Collective fault 18 IND18
Unacknowledged fault 19 IND19
20 IND110
X2 alarm 21 IND111
The group is in operation 22 IND112
Sequence in progress 23 IND113
24 IND114
Holding active 25 IND115
Interlock IB1, IB2 block 30 IND20
LOCAL point of control 31 IND21
TEST point of control 32 IND22
STANDBY point of control 33 IND23
SEQUENCE point of control 34 IND24
AUTO/MAN control mode 35 IND25
Ready for start 36 IND26
Ready for stop 37 IND27
IC1 38 IND28
IC2 39 IND29
IB1 40 IND210
IB2 41 IND211
IB3 42 IND212
IB4 43 IND213
IA 44 IND214
Block interlocking active 45 IND215
70 RRES
Active step number 70 IRES

(1) The picture is unique for this manual. It is not included in the manual *PC Elements Advant Controller 400 Series Reference Manual*, which instead contains a basic, simplified picture of the PC element MMC-IND.

MMC-IND with C1 and C2 specified as 16 to obtain a sufficient number of binary inputs for IND1 and IND2. C3 and C4 are not utilized.

Call MMC-IND (C1, C2, C3, C4)

Parameter	Significance	Permissible values
C1	Number of inputs for IND1 indications.	0 .. 16
C2	Number of inputs for IND2 indications.	0 .. 16
C3(1)	Inputs used for IND1. C3=0: Indications via inputs 10 - 25. C3=1: Indications via input 60.	0 .. 1
C4(2)	Inputs used for IND2. C4=0: Indications via inputs 30 - 45. C4=1: Indications via input 61.	0 .. 1

(1) If C3 is set to 1, C1 must be 0.
(2) If C3 is set to 1, C1 must be 0.

Terminal Description(1)

No	Name	Type	Description
1	PCPAR2	IA10	**PC PAR**ameter **2**. Parameters from the MMCX data base element. *Parameters from the MMCX data base element. Connection to terminal PC_PARAM2 at the element concerned in MMCX. A group of signals which contain parameters for updating of MMCX.*
5	PCRES	OA12	**PC RES**ult. Result to the MMCX data base element. *Result to MMCX data base element. Connection to the terminal PC_RES at the element concerned in MMCX. A group of signals which contain the control signals at the PC element inputs.*
10 – 45	*IND10 – IND215*	*IB*	Connections for the user-defined PC circuits. Indications from the user-defined PC program. These signals can be supervised with regard to alarms and events. An example is described below.
10	IND10	IB	**IND**ication **1** input **0**. Input for the first indication within IND1. *Input for the first indication within IND1. Timeout, sequence execution. The total time for startup or shutdown of the group is too long.*
11	IND11	IB	**IND**ication **1** input **1**. Input for the second indication within IND1. *Input for the second indication within IND1. Timeout, step execution. The total time for execution of one of the steps is too long.*
12 – 14	*IND12– IND14*	*IB*	*Fault 3 – 5. Undefined fault indication.*
15 – 16	*IND15– IND16*	*IB*	*Not used.*
17	*IND17*	*IB*	*X1 alarm. Undefined fault indication. Can be defined with text in displays (PRESA).*
18	*IND18*	*IB*	*Pilot alarm signal. At least one of the alarms are persistent.*
19	*IND19*	*IB*	*Unacknowledged alarm. At least one of the alarms are unacknowledged.*
20	*IND110*	*IB*	*Not used.*
21	*IND111*	*IB*	*X2 alarm. Undefined fault indication. Can be defined with text in displays (PRESB).*
22	*IND112*	*IB*	*The group is in operation. Acknowledgement of the entire group being in operation.*

Terminal Description(1)

No	Name	Type	Description
23	*IND113*	*IB*	*The sequence is running. E.g. the output RUN on the sequence header.*
24	*IND114*	*IB*	*Not used.*
25	*IND115*	*IB*	*The sequence is held (HOLD). E.g. the output HOLD on the sequence header.*
30	IND20	IB	**IND**ication **2** input **0**. Input for the first indication within IND2. *Input for the first indication within IND2. The interlocks IB1 and IB2 are disabled, i.e. disconnected.*
31	IND21	IB	**IND**ication **2** input **1**. Input for the second indication within IND2. *Input for the second indication within IND2. The mode of control LOCAL is invoked.*
32	*IND22*	*IB*	*The mode of control TEST is invoked.*
33	*IND23*	*IB*	*The mode of control STAND BY is invoked.*
34	*IND24*	*IB*	*The mode of control SEQUENCE is invoked.*
35	*IND25*	*IB*	*The modes of control AUTO/MAN. AUTO applies when the signal is set at 1. Otherwise, MANUAL applies.*
36	*IND26*	*IB*	*Ready for start. The group is ready for start.*
37	*IND27*	*IB*	*Ready for stop. The group is ready for stop.*
38 – 39	*IND28– IND29*	*IB*	*IC1 – IC2. Interlocks preventing the sequence from starting.*
40 – 44	*IND210– IND214*	*IB*	*IB1 – IA. Interlocks preventing the sequence from starting.*
45	*IND215*	*IB*	*Interlock signal is active but d isabled. Any of the interlocks 1 B 1 or 1 B2 are active but prevented from stopping the sequence.*
60	IND1	II	**IND**ication **1**. Input for IND1 indications in packed form.
61	IND2	II	**IND**ication **2**. Input for IND2 indications in packed form.
70	RRES	IR	**R**eal **RES**ult. Input for real measured value. *Input for real measured value.*
71	IRES	IIL	**I**nteger **RES**ult. Input for integer measured value. *Input for measured integer value. Integer indicating the currently active step of the group start sequence.*

(1) The texts in italic are unique for this manual. They are not included in the manual *PC Elements Advant Controller 400 Series Reference Manual*.

The signals are continuous, i.e. the indications, e.g. on the displays in the operator station, remain unchanged as long as the signals remain in the one or the other position.

The functions of the different signals are described along with the display elements for GROUP. For this reason, only a brief summary is given here.

4.3.2 PC Element MMC-ORD

The figure below shows how the control signals are distributed to the different terminals on the PC elements MMC-ORD for the functional unit GROUP. This is followed by a short supplementax explanation of the use of the element in connection with the functional unit GROUP. See also the preceding Section 4.2 Function.

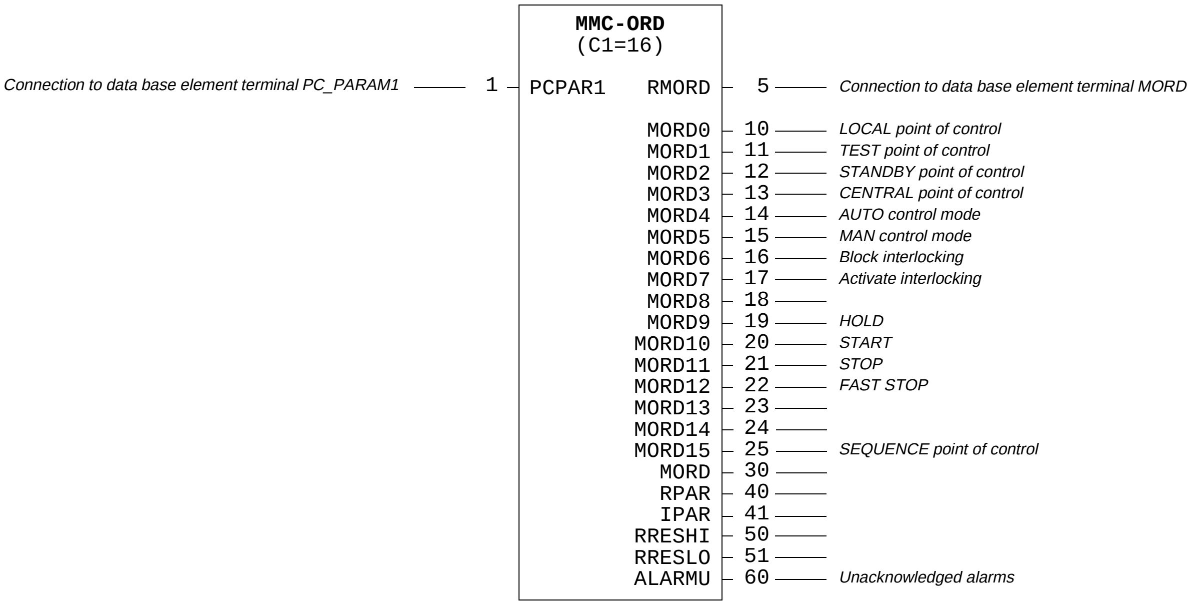
Operator Station Orders MMC-ORD

Summary

MMC-ORD is used with the data base element MMCX, object handlers and a suitable operator dialogue to give the PC programmer the possibilities to send orders, set point values, limit values and information of unacknowledged alarms from the operator to the PC program. The element can handle 16 on/off orders, one real set point value and one integer set point value.

If the element is used with an MMC-IND element in the PC program for a certain object, it also has the capability of presenting information of whether unacknowledged alarms are present among the 32 indications of the MMC-IND element or not. MMC-ORD can also present high and low limits for the real measured value of the MMC-IND element.

Picture Operation Station Orders element (1)


MMC-ORD
(C1=16)
Connection to data base element terminal PC_PARAM1
1
PCPAR1
RMORD
5
Connection to data base element terminal MORD
MORD0 10 LOCAL point of control
MORD1 11 TEST point of control
MORD2 12 STANDBY point of control
MORD3 13 CENTRAL point of control
MORD4 14 AUTO control mode
MORD5 15 MAN control mode
MORD6 16 Block interlocking
MORD7 17 Activate interlocking
MORD8 18
MORD9 19 HOLD
MORD10 20 START
MORD11 21 STOP
MORD12 22 FAST STOP
MORD13 23
MORD14 24
MORD15 25 SEQUENCE point of control
MORD 30
RPAR 40
IPAR 41
RRESHI 50
RRESLO 51
ALARMU 60 Unacknowledged alarms

---

(1) The picture is unique for this manual. It is not included in the manual *PC Elements Advant Controller 400 Series Reference Manual*, which instead contains a basic, simplified picture of the PC element MMC-ORD.

MMC-ORD with C1 specified as 16 to obtain a sufficient number of binary commands.

Call MMC-ORD (C1)

Parameter	Significance	Permissible values
C1	Number of on/off orders	1 .. 16

Terminal Description[1]

No	Name	Type	Description
1	PCPAR1	IA24	**PC PAR**ameter **1**. Parameters from the MMCX data base element. *Connection to the terminal PC_PARAM 1 data base element. A group of signals which contains all of the commands which are distributed to the outputs of the element.*
5	RMORD	OI	**R**eset **M**anual **ORD**er. Output for clearing on/off orders. *Connection to the MORD data base element. A group of signals which contains some of the control signals which are included in PCPAR1. The signals are to acknowledge that the PC element has reacted as intended to the commands given by MMCX by resetting them in MMCX and thereby permitting the operator to give further commands.*
10	MORD0	OB	**M**anual **ORD 0**. Output for the 1:st on/off order. *Output for the 1:st on/off order. Connections to the user-defined PC circuit. All are binary and are true for the orders given during one execution in accordance with the preceding RMORD function. Their control function is described in the preceding section Function.*
11 : 10+C1–1	MORD1 MORD(*C1–1*)	OB OB	**M**anual **ORD**er **1**. Output for the 2:nd on/off order. : **M**anual **ORD**er C1–1. Output for the (*C1–1*):st on/off order.
30	MORD	OI	**M**anual **ORD**er. Output for the on/off orders in packed form. *Output for the on/off orders in packed form.*
40	RPAR	OR	**R**eal **PAR**ameter. Output for the real set point value.
41	IPAR	OIL	**I**nteger **PAR**ameter. Output for the integer set point value (2).
50	RRESHI	OR	**R**eal **RES**ult **HI**gh limit. High limit for the real measured value (2).
51	RRESLO	OR	**R**eal **RES**ult **LO**w limit. Low limit for the real measured value (2).
60	ALARMU	OIL	**ALARM U**nacknowledged. Output for information about unacknowledged alarms among the indications. *Corresponding to IND10–IND215.*

(1) The texts in italic are unique for this manual. They are not included in the manual *PC Elements Advant Controller 400 Series Reference Manual.*

(2) Applies to the companion MMC-IND element.

A PC element of CONV-IB (IL, 4, 31) type connected to ALARMU will indicate the non-acknowledgement for IND1 and IND2.

Indication (1)	**CONV-IB terminal**
IND10	O1
IND11	O2
IND12	O3
IND13	O4
IND14	O5
IND15	O6
IND16	O7
IND17	O8
IND18	O9
IND19	O10
IND110	O11
IND111	O12
IND112	O13
IND113	O14
IND114	O15
IND115	O16
IND20	O17
IND21	O18
IND22	O19
IND23	O20
IND24	O21
IND25	O22
IND26	O23
IND27	O24
IND228	O25
IND29	O26
IND210	O27
IND211	O28
IND212	O29
IND213	O30
IND214	O31
IND215	SIGN

(1) IND10 – IND215 refers to terminals on the PC element MMC-IND.

4.4 Data Base Element MMCX

The data base element MMCX is used for the functional unit GROUP. It is important to note that the value for Reference type (RTYPE) is 39 to obtain the dialogs and the object display which is intended for the functional unit GROUP.

The figures below shows which of the properties in the data base it is necessary to consider when using the functional unit GROUP. The original values for MMCX data base elements are specified. Note that the complete element is not presented.

The figure is followed by an explanation of the different properties and a specification of their values in relation to the application of GROUP.

The general data base element MMCX is described in the manual *Data Base Elements Advant Controller 400 Series Reference Manual*.

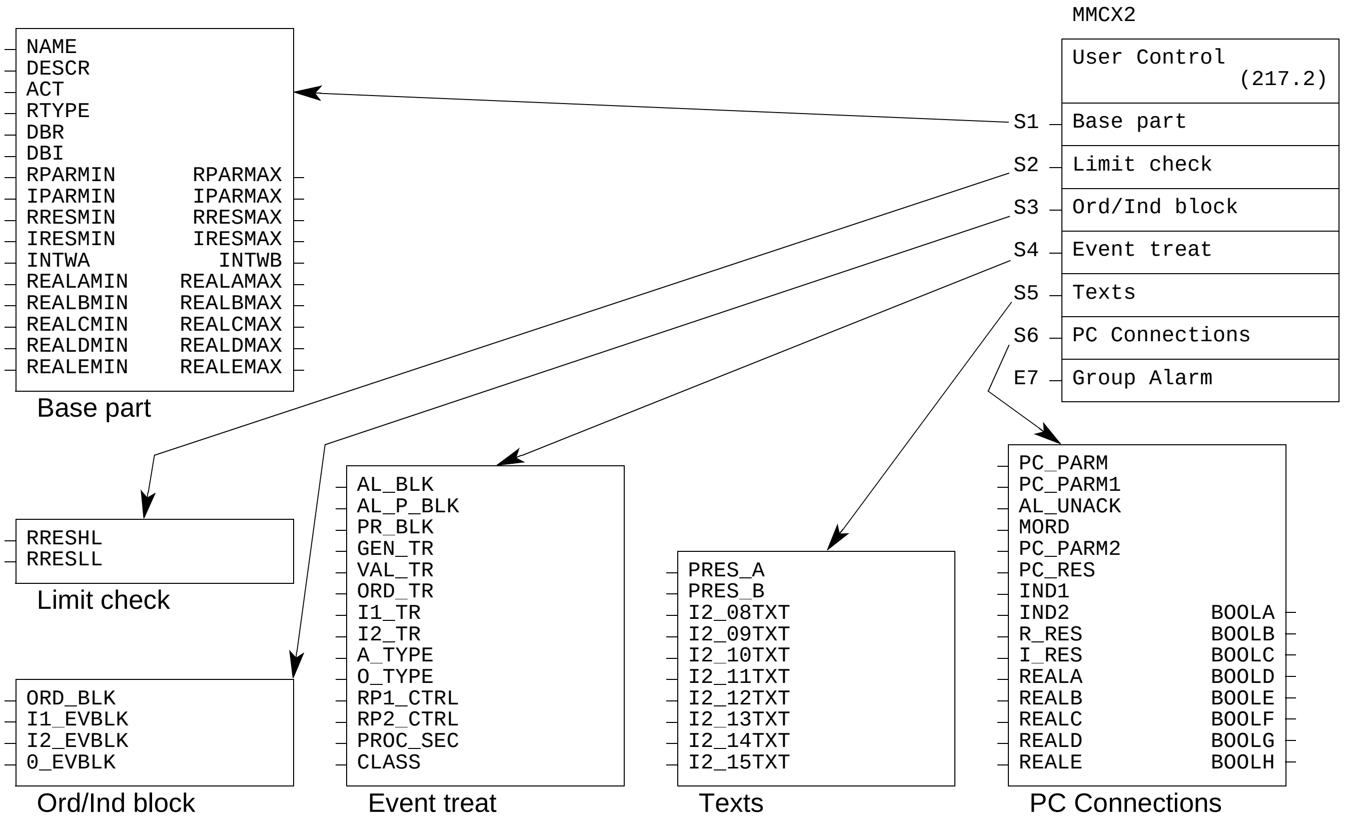
User Control MMCX

Summary

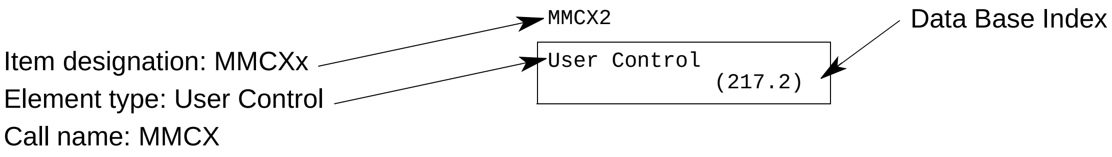
The User Control data base element is used to build Advant Station 500 Series Operator Station functions for the MMCX function unit. Some static information is given. For information about the extension parts of Group Alarm see data base element Group Member.

Parameters that can be changed from the dialog can be given start values.

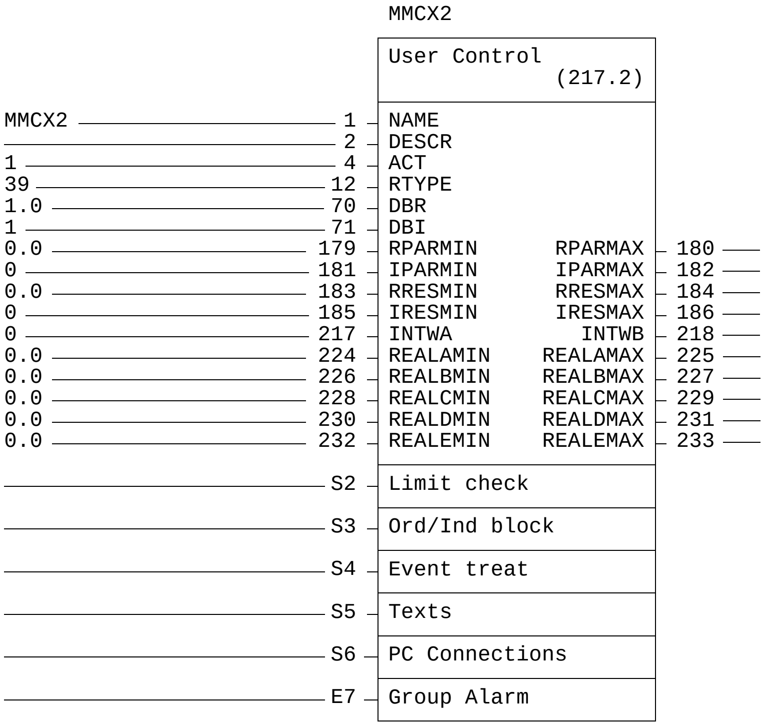
Overview



Head



Base part


MMCX2
User Control
(217.2)
MMCX2 1 NAME
2 DESCR
1 4 ACT
39 12 RTYPE
1.0 70 DBR
1 71 DBI
0.0 179 RPARMIN RPARMAX 180
0 181 IPARMIN IPARMAX 182
0.0 183 RRESMIN RRESMAX 184
0 185 IRESMIN IRESMAX 186
0 217 INTWA INTWB 218
0.0 224 REALAMIN REALAMAX 225
0.0 226 REALBMIN REALBMAX 227
0.0 228 REALCMIN REALCMAX 229
0.0 230 REALDMIN REALDMAX 231
0.0 232 REALEMIN REALEMAX 233
S2 Limit check
S3 Ord/Ind block
S4 Event treat
S5 Texts
S6 PC Connections
E7 Group Alarm

Terminals User Control elements, Base part (1)

Terminal No	Terminal Name	Value entered by	Default value	PC connection data type	Description	Remarks
1	NAME	user	MMCXx	-	Each object must be given a unique **NAME**. E.g. ABC 123.	
					Instance name of the group start. Used, for example, by the operator in selecting an object with the help of its name. Max 20 characters.	
2	DESCR	user	–	–	**DESCR**iption of the object that is shown in Operator Station.	Max. 20 characters. Any information regarding the object may be entered.
					Additional description of the group start. Appears together with NAME in most standard displays.	
4	ACT	user	1	B(r/w)	1=operator communication is **ACT**ive.	0=communication is blocked and events are not updated.
					When ACT is set high, it is possible for the operator to communicate with the group at hand.	

Terminals User Control elements, Base part (1)

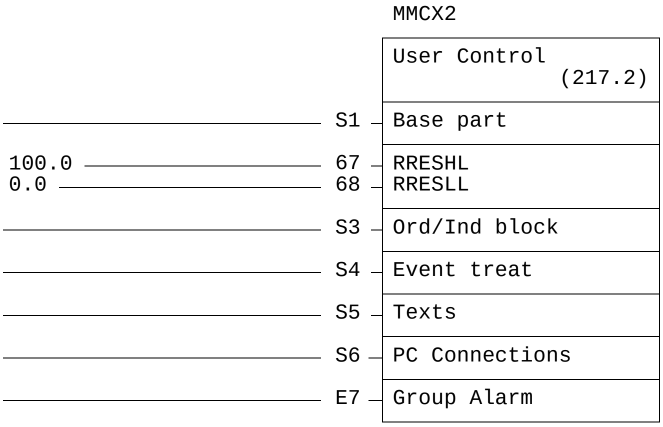
Terminal No	Terminal Name	Value entered by	Default value	PC connection data type	Description	Remarks
12	RTYPE	user	39	–	**R**eference **TYPE** for GROUP.	(32-39)
					The parameter should be set at 39, since this number provides access to the correct dialog and object display for the group start.	
70	DBR	user	1.0	R(r/w)	Delta **R**eal determines the size of the change in value of REAL RES that gives event controlled updating of the picture.	–
71	DBI	user	1	IL(r/w)	Delta **I**ntl determines the size of the change in value of INTL RES that gives event controlled updating of the picture.	–
					When DBI is set = 1, the presented display is updated by each new step.	
179	RPARMIN	user	0.0	R(r/w)	**R**eal **PAR**ameter **MIN**imum. Minimum value of REAL PARAM.	–
180	RPARMAX	user	100.0	R(r/w)	**R**eal **PAR**ameter **MAX**imum. Maximum value of REAL PARAM.	–
181	IPARMIN	user	0	IL(r/w)	**I**ntl **PAR**ameter **MIN**imum. Minimum value of INTL PARAM.	–
182	IPARMAX	user	100	IL(r/w)	**I**ntl **PAR**ameter **MAX**imum. Maximum value of INTL PARAM.	–
183	RRESMIN	user	0.0	R(r/w)	**R**eal **RES MIN**imum. Minimum value of REAL RES.	–
184	RRESMAX	user	100.0	R(r/w)	**R**eal **RES MAX**imum. Maximum value of REAL RES.	–
185	IRESMIN	user	0	IL(r/w)	**I**ntl **RES MIN**imum. Minimum value of INTL RES.	–
					0 and 100 states that the least and maximum numerical values for the step number are 0 and 100 respectively. Used for plausibility check of the numerical value.	
186	IRESMAX	user	100	IL(r/w)	**I**ntl **RES MAX**imum. Maximum value of INTL RES.	–
					0 and 100 states that the least and maximum numerical values for the step number are 0 and 100 respectively. Used for plausibility check of the numerical value.	

Terminals User Control elements, Base part (1)

<table>
<tr><th>Termi-nal No</th><th>Terminal Name</th><th>Value entered by</th><th>Default value</th><th>PC con-nection data type</th><th>Description</th><th>Remarks</th></tr>
<tr><td rowspan="2">217</td><td rowspan="2">INTWA</td><td rowspan="2">user</td><td rowspan="2">0</td><td rowspan="2">I(r/w)</td><td>INTeger Word A. Integer word to be used by PC program.</td><td>–</td></tr>
<tr><td colspan="2"><i>The value B0000 0000 0000 0000 indicates that all of | the texts in the object display will be seen.
The following texts can be removed by means of INTWA bit pattern when the value is set to 1.</i>
<table>
<tr><th><i>Text</i></th><th><i>Bit value</i></th></tr>
<tr><td><i>Long operating time seq.</i></td><td><i>1 (LSB, the right hand)</i></td></tr>
<tr><td><i>Long operating time step</i></td><td><i>2</i></td></tr>
<tr><td><i>Fault 3</i></td><td><i>4</i></td></tr>
<tr><td><i>Fault 4</i></td><td><i>8</i></td></tr>
<tr><td><i>Fault 5</i></td><td><i>1</i></td></tr>
<tr><td><i>Not used</i></td><td><i>2</i></td></tr>
<tr><td><i>“</i></td><td><i>4</i></td></tr>
<tr><td><i>“</i></td><td><i>.</i></td></tr>
<tr><td><i>“</i></td><td><i>.</i></td></tr>
<tr><td><i>“</i></td><td><i>8 (MSB, the left hand)</i></td></tr>
</table></td></tr>
<tr><td>218</td><td>INTWB</td><td>user</td><td>0</td><td>I(r/w)</td><td>INTeger Word B. Integer word to be used by PC program.</td><td>–</td></tr>
<tr><td>224</td><td>REALAMIN</td><td>user</td><td>0.0</td><td>R(r)</td><td>MINimum value of REAL A.</td><td>–</td></tr>
<tr><td>225</td><td>REALAMAX</td><td>user</td><td>100.0</td><td>R(r)</td><td>MAXimum value of REAL A.</td><td>–</td></tr>
<tr><td>226</td><td>REALBMIN</td><td>user</td><td>0.0</td><td>R(r)</td><td>MINimum value of REAL B.</td><td>–</td></tr>
<tr><td>227</td><td>REALBMAX</td><td>user</td><td>100.0</td><td>R(r)</td><td>MAXimum value of REAL B.</td><td>–</td></tr>
<tr><td>228</td><td>REALCMIN</td><td>user</td><td>0.0</td><td>R(r)</td><td>MINimum value of REAL C.</td><td>–</td></tr>
<tr><td>229</td><td>REALCMAX</td><td>user</td><td>100.0</td><td>R(r)</td><td>MAXimum value of REAL C.</td><td>–</td></tr>
<tr><td>230</td><td>REALDMIN</td><td>user</td><td>0.0</td><td>R(r)</td><td>MINimum value of REAL D.</td><td>–</td></tr>
<tr><td>231</td><td>REALDMAX</td><td>user</td><td>100.0</td><td>R(r)</td><td>MAXimum value of REAL D.</td><td>–</td></tr>
<tr><td>232</td><td>REALEMIN</td><td>user</td><td>0.0</td><td>R(r)</td><td>MINimum value of REAL E.</td><td>–</td></tr>
<tr><td>233</td><td>REALEMAX</td><td>user</td><td>100.0</td><td>R(r)</td><td>MAXimum value of REAL E.</td><td>–</td></tr>
</table>

(1) The texts in italic in the column “Description” and “Remarks” are unique for this manual. They are not included in the manual *Data Base Elements Advant Controller 400 Series Reference Manual*.

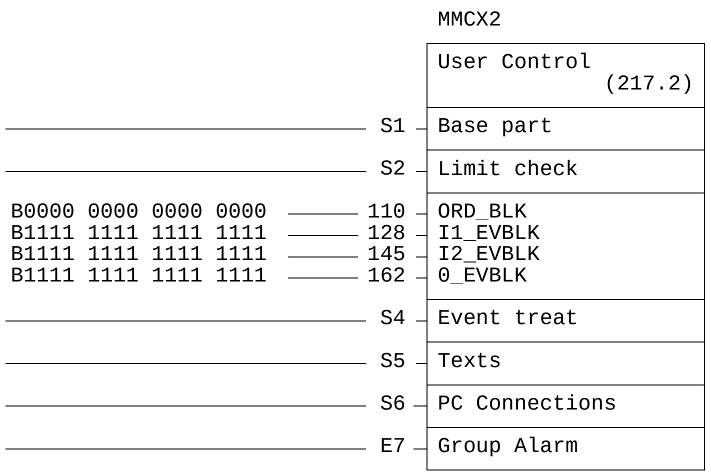
Limit check



Terminals User Control element, Limit check

Terminal No	Terminal Name	Value entered by	Default value	PC connection data type	Descriptions	Remarks
67	RRESHL	user	100.0	R(r/w)	**R**eal **RES H**igh **L**imit. Highest limit used by PC for limit check of the real value REAL RES.	–
68	RRSLL	user	0.0	R(r/w)	**R**eal **RES L**ow **L**imit. Lowest limit used by PC for limit check of the real value REAL RES.	–

Ord/Ind block



Terminals, Ord/Ind block (1)

Terminal No	Terminal Name	Value entered by	Default value	PC connection data type	Description	Remarks
110	ORD_BLK	user	H’0000	I(r/w)	**BL**oc**K**ing of manual **ORD**ers. New value can be entered as a hexadecimal string, e.g. H’000, or as a binary string, e.g. B0000 0000 0000 0000.	–
					With the value B0000 0000 0000 0000, all commands which the dialogs can give reach MMC-ORD. Commands are blocked by setting certain bits to 1 in accordance with the pattern below. *Command – Bit value* *Local – 1 (LSB, the right hand)* *Test – 2* *Standby – 4* *Central – 8* *Auto – 1* *Man – 2* *Block of interlocking – 4* *Activate interlocking – 8* *Not used – 1* *Hold – 2* *Start – 4* *Stop – 8* *Fast stop – 1* *Not used – 2* *Not used – 4* *SEQUENCE point of control – 8 (MSB, the left hand)*	

Terminals, Ord/Ind block (Continued) (1)

<table>
<tr><th>Terminal No</th><th>Terminal Name</th><th>Value entered by</th><th>Default value</th><th>PC connection data type</th><th>Description</th><th>Remarks</th></tr>
<tr><td rowspan="2">128</td><td rowspan="2">I1_EVBLK</td><td rowspan="2">user</td><td rowspan="2">H'FFFF</td><td rowspan="2">I(r/w)</td><td>BLocKing of Indication EVents 1. New value can be entered as a hexadecimal string, eg. H'0000, or as a binary string, e.g. B0000 0000 0000 0000.</td><td>–</td></tr>
<tr><td colspan="2"><i>The value B 1111 1111 1111 1111 means that no signal will be event-treated by the Operator Station. Event-treatment is created by setting certain bits to 0 in accordance with the following pattern.</i>
<table>
<tr><th><i><u>Description</u></i></th><th><i><u>Bit value</u></i></th></tr>
<tr><td><i>Long operating time seq.</i></td><td><i>1 (LSB, the right hand)</i></td></tr>
<tr><td><i>Long operating time step</i></td><td><i>2</i></td></tr>
<tr><td><i>Fault 3</i></td><td><i>4</i></td></tr>
<tr><td><i>Fault 4</i></td><td><i>8</i></td></tr>
<tr><td><i>Fault 5</i></td><td><i>1</i></td></tr>
<tr><td><i>Not used</i></td><td><i>2</i></td></tr>
<tr><td><i>Not used</i></td><td><i>4</i></td></tr>
<tr><td><i>X1 alarm</i></td><td><i>8</i></td></tr>
<tr><td><i>Collective fault</i></td><td><i>1</i></td></tr>
<tr><td><i>Unacknowledged fault</i></td><td><i>2</i></td></tr>
<tr><td><i>Not used</i></td><td><i>4</i></td></tr>
<tr><td><i>X2 alarm</i></td><td><i>8</i></td></tr>
<tr><td><i>The group in operation</i></td><td><i>1</i></td></tr>
<tr><td><i>Sequence in progress</i></td><td><i>2</i></td></tr>
<tr><td><i>Not used</i></td><td><i>4</i></td></tr>
<tr><td><i>Holding active</i></td><td><i>8 (MSB, the left hand)</i></td></tr>
</table></td></tr>
</table>

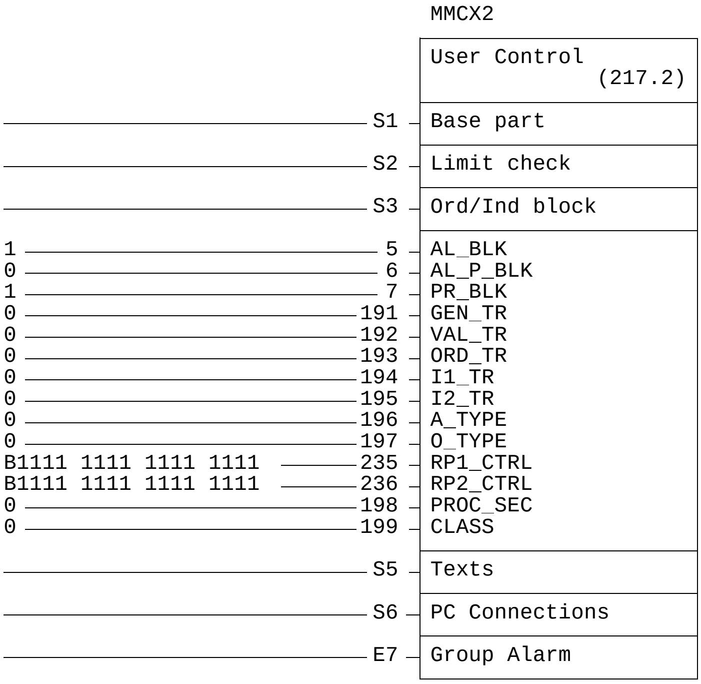
Terminals, Ord/Ind block (Continued) (1)

<table>
<thead>
<tr><th>Terminal No</th><th>Terminal Name</th><th>Value entered by</th><th>Default value</th><th>PC connection data type</th><th>Description</th><th>Remarks</th></tr>
</thead>
<tbody>
<tr><td rowspan="19">145</td><td rowspan="19">I2_EVBLK</td><td rowspan="19">user</td><td rowspan="19">H'FFFF</td><td rowspan="19">I(r/w)</td><td>BLocKing of Indication EVents 2. New value can be entered as a hexadecimal string, eg. H'0000, or as a binary string, e.g. B0000 0000 0000 0000.</td><td>–</td></tr>
<tr><td colspan="2"><i>The value B 1111 1111 1111 1111 means that no signal will be event treated by the Operator Station. Event-treatment is created by setting certain bits to 0 in accordance with the pattern below.</i></td></tr>
<tr><td><i><u>Indication</u></i></td><td><i><u>Bit value</u></i></td></tr>
<tr><td><i>Interlock IB1, IB2 block</i></td><td><i>1 (LSB, the right hand)</i></td></tr>
<tr><td><i>Local</i></td><td><i>2</i></td></tr>
<tr><td><i>Test</i></td><td><i>4</i></td></tr>
<tr><td><i>Standby point of control</i></td><td><i>8</i></td></tr>
<tr><td><i>Sequence point of control</i></td><td><i>1</i></td></tr>
<tr><td><i>Auto/Man control mode</i></td><td><i>2</i></td></tr>
<tr><td><i>Ready for start</i></td><td><i>4</i></td></tr>
<tr><td><i>Ready for stop</i></td><td><i>8</i></td></tr>
<tr><td><i>IC1</i></td><td><i>1</i></td></tr>
<tr><td><i>IC2</i></td><td><i>2</i></td></tr>
<tr><td><i>IB1</i></td><td><i>4</i></td></tr>
<tr><td><i>IB2</i></td><td><i>8</i></td></tr>
<tr><td><i>IB3</i></td><td><i>1</i></td></tr>
<tr><td><i>IB4</i></td><td><i>2</i></td></tr>
<tr><td><i>IA</i></td><td><i>4</i></td></tr>
<tr><td><i>Block interlocking active</i></td><td><i>8 (MSB, the left hand)</i></td></tr>
<tr><td rowspan="2">162</td><td rowspan="2">O_EVBLK</td><td rowspan="2">user</td><td rowspan="2">H'FFFF</td><td rowspan="2">–</td><td>BLocKing of EVents at manual Orders. New value can be entered as a hexadecimal string, e.g. H'0000, or as a binary string,e.g. B0000 0000 0000 0000.</td><td>–</td></tr>
<tr><td colspan="2"><i>Commands cannot be event-treated in the functional unit GROUP. The value is set to B1111 111111111111.</i></td></tr>
</tbody>
</table>

---

(1) The texts in italic in the column “Description” and “Remarks” are unique for this manual. They are not included in the manual *Data Base Elements Advant Controller 400 Series Reference Manual*.

Event treat



Terminals, Event treat (1)

Terminal No	Terminal Name	Value entered by	Default value	PC connection data type	Description	Remarks
5	AL_BLK	dialog	1	–	**BL**ocKing of **AL**arms to alarm list.	–
					Gives blocking of alarm to the alarm list. The printer and the top line in displays on the screen function however. Blocking and deblocking are performed by the operator.	
6	AL_P_BLK	system	0	B(r/w)	**AL**arm **P**eriod **BL**oc**K** flag for temporary blocking of alarm indication and event handling in Operator Station by PC program.	–
					There is no alarm and event-treatment if the value is 1. The property is intended to be controlled by PC.	
7	PR_BLK	dialog	1	–	**BL**oc**K**ing of alarms to **PR**inter and other units.	–
					Blocking of alarm to printer and other units if the value is 1. The operator performs blocking and deblocking. Note that if both AL_BLK and PR_BLK are 1 the effect is the same as if AL_P_BLK is 1.	

Terminals, Event treat (Continued) (1)

Terminal No	Terminal Name	Value entered by	Default value	PC connection data type	Description	Remarks
191	GEN_TR	user	0	–	The **GEN**eral **TR**eatment terminal assigns item numbers for event handling of ALARM BLocK orders and PRINT BLocK orders.	–
					Not used.	
192	VAL_TR	user	0	–	**VAL**ue **TR**eatment. Item numbers for event treatment of REAL PARAM orders, INTL PARAM orders, REAL RES LOLIM orders and REAL RES HILIM orders.	–
193	ORD_TR	user	0	–	The manual **ORD**er **TR**eatment terminal assigns item numbers for event handling of manual orders MORD_00..MORD_15.	–
					Not used.	
194	I1_TR	user	0	–	The **I**ndication **TR**eatment of IND**1** terminal specifies which Event Treat data base element is to be used in the Operator Station.	–
					The value specifies the record number of the EVENT TREAT file in the Operator Station. This describes how the signals which are covered by this pointer are to be alarm and event-treated. See Section 4.5.3 Event Handling.	
195	I2_TR	user	0	–	The **I**ndication **TR**eatment of IND**2** terminal specifies which EVENT TREAT data base element is to be used in the Operator Station.	–
					The value indicates the record number of the EVENT TREAT file in the Operator Station. This describes how the signals covered by this pointer are to be alarm and event treated. See Section 4.5.3 Event Handling.	
196	A_TYPE	user	0	–	**A**nalyse **TYPE**. 0=Error handling of IND1 and IND2 in Operator Station 1=No error handling of IND1 or IND2 except for IND1_08.	–
197	O_TYPE	user	0	–	**O**rder **TYPE**.	Always set to 0.

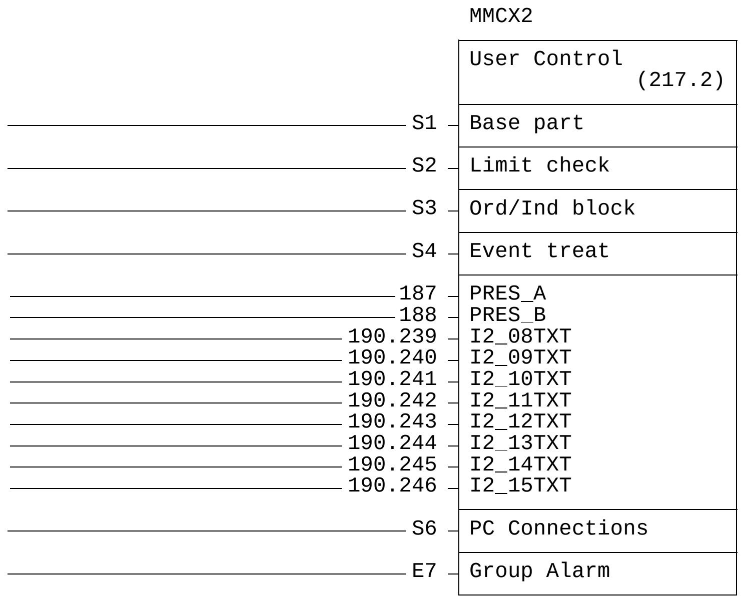
Terminals, Event treat (Continued) (1)

Terminal No	Terminal Name	Value entered by	Default value	PC connection data type	Description	Remarks
235	RP1_CTRL	user	H’FFFF	–	Every bit of the value **R**e**P**eat **C**on**TR**o**L 1** blocks error in the corresponding bit of the IND1. New value can be entered as a hexadecimal string, e.g. H'0000, or as a binary string, e.g. B0000 0000 0000 0000.	–
					Controls the alarm handling for signals in IND1. See the previous pattern for IND1 signals (11_EVBLK). The value B1111 1111 1111 1111 prevents the repetition of faults (flutter) filling the alarm list. Only two faults are included in the list and more are added after acknowledgement of the alarm.	
236	RP2_CTRL	user	H'FFFF	–	Every bit of the value **R**e**P**eat **C**on**TR**o**L 2** blocks error in the corresponding bit of the IND2. New value can be entered as a hexadecimal string, e.g. H'0000, or as a binary string, e.g. B0000 0000 0000 0000.	–
					Controls the alarm handling for signals in IND2. See previous pattern for IND2 signals (12_EVBLK). The value B1111 1111 1111 1111 prevents the repetition of faults (flutter) filling the alarm list. Only two faults are included in the list and more faults are added after acknowledgement of the alarm.	
198	PROC_SEC	user	0	I(r/w)	**PROC**ess **SEC**tion is used for sectioning the alarm handling. Section numbers 1-16. 0=no sectioning. The signal may be presented in lists –1=the signal cannot be selected and cannot be presented in lists.	–
199	CLASS	user	0	I(r/w)	**CLASS** subdivides process section. Used by status list function in Operator Station.	–

---

(1) The texts in italic in the column “Description” and “Remarks” are unique for this manual. They are not included in the manual *Data Base Elements Advant Controller 400 Series Reference Manual*.

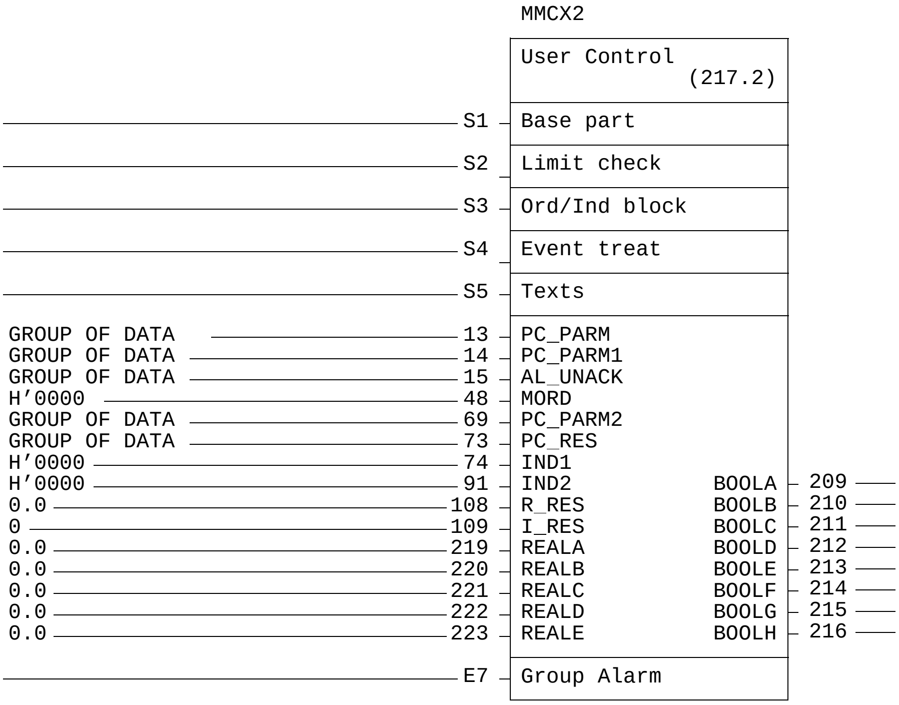
Texts



Terminals, Texts

Terminal No	Terminal Name	Value entered by	Default value	PC connection data type	Description	Remarks
187	PRES_A	user	–	–	**PRES A**. Freely usable text.	Max. 10 characters.
188	PRES_B	user	–	–	**PRES B**. Freely usable text.	Max. 10 characters.
190.239	I2_08TXT	user	–	–	**I**nd**2 08** pres **T**e**XT**. Text attached to IND2 08.	Highest priority. Max. 17 characters.
190.240	I2_09TXT	user	–	–	**I**nd**2 09** pres **T**e**XT**. Text attached to IND2 09.	Max. 17 characters.
190.241	I2_10TXT	user	–	–	**I**nd**2 10** pres **T**e**XT**. Text attached to IND2 10.	Max. 17 characters.
190.242	I2_11TXT	user	–	–	**I**nd**2 11** pres **T**e**XT**. Text attached to IND2 11.	Max. 17 characters.
190.243	I2_12TXT	user	–	–	**I**nd**2 12** pres **T**e**XT**. Text attached to IND2 12.	Max. 17 characters.
190.244	I2_13TXT	user	–	–	**I**nd**2 13** pres **T**e**XT**. Text attached to IND2 13.	Max. 17 characters.
190.245	I2_14TXT	user	–	–	**I**nd**2 14** pres **T**e**XT**. Text attached to IND2 14.	Max. 17 characters.
190.246	2_15TXT	user	–	–	**I**nd**2 15** pres **T**e**XT**. Text attached to IND2 15.	Lowest priority. Max. 17 characters.

PC Connections



Terminals, PC Connections

Termi-nal No	Terminal Name	Value entered by	Default value	PC con-nection data type	Description	Remarks
13	PC_PARM	system	–	A32(r/w)	The **PC PAR**a**M**eters terminal contains group data to be connected to the PARAM terminals of the VALVECON and MOTCON PC elements.	–
14	PC_PARM1	system	–	A24(r/w)	Group data to be connected to the **PCPAR1** terminal of the MMC-ORD PC element.	–
15	AL_UNACK	system	–	IL(r/w)	Group data over **UNACK**nowledged **AL**arms.	–
48	MORD	dialog and PC	–	I(r/w)	Group data for **M**anual **ORD**ers from the operator. Used for acknowledge from PC.	–
69	PC_PARM2	dialog and PC	–	A10(r/w)	Group data to be connected to the **PCPAR2** terminal of the MMC-IND PC element.	–

Terminals, PC Connections (Continued)

Terminal No	Terminal Name	Value entered by	Default value	PC connection data type	Description	Remarks
73	PC_RES	PC	–	A12(r/w)	The **PC RES**ult terminal contains group data to be connected from the RESULT terminal of the VALVECON and MOTCON PC elements and to MMC-IND terminal PCRES.	–
74	IND1	system	–	I(r/w)	The **IND**ication **1** terminal contains status indications from PC.	–
91	IND2	system	–	I(r/w)	The **IND**ication **2** terminal contains status indications from PC.	–
108	R_RES	system	–	R(r/w)	**R**eal **RES**ult from PC.	–
109	I_RES	system	–	IL(r/w)	Long **I**nteger **RES**ult from PC.	–
219	REALA	PC	0.0	R(r/w)	**REAL A**. Extra real values to be used for presentation by PC program.	–
220	REALB	PC	0.0	R(r/w)	**REAL B**. Extra real values to be used for presentation by PC program.	–
221	REALC	PC	0.0	R(r/w)	**REAL C**. Extra real values to be used for presentation by PC program.	–
222	REALD	PC	0.0	R(r/w)	**REAL D**. Extra real values to be used for presentation by PC program.	–
223	REALE	PC	0.0	R(r/w)	**REAL E**. Extra real values to be used for presentation by PC program.	–
209	BOOLA	PC	0	B(r/w)	**BOOL**ean **A**. Extra free flags to be used by PC program.	–
210	BOOLB	PC	0	B(r/w)	**BOOL**ean **B**. Extra free flags to be used by PC program.	–
211	BOOLC	PC	0	B(r/w)	**BOOL**ean **C**. Extra free flags to be used by PC program.	–
212	BOOLD	PC	0	B(r/w)	**BOOL**ean **D**. Extra free flags to be used by PC program.	–
213	BOOLE	PC	0	B(r/w)	**BOOL**ean **E**. Extra free flags to be used by PC program.	–
214	BOOLF	PC	0	B(r/w)	**BOOL**ean **F**. Extra free flags to be used by PC program.	–
215	BOOLG	PC	0	B(r/w)	**BOOL**ean **G**. Extra free flags to be used by PC program.	–
216	BOOLH	PC	0	B(r/w)	**BOOL**ean **H**. Extra free flags to be used by PC program.	–

4.5 Operator Station Functions

The Operator functions are divided in principle into 4 parts:

- Presentation
- Dialog
- Event treatment
- Group alarm

4.5.1 Presentation

Display elements which can be used for different display types are available for use in the functional unit GROUP.

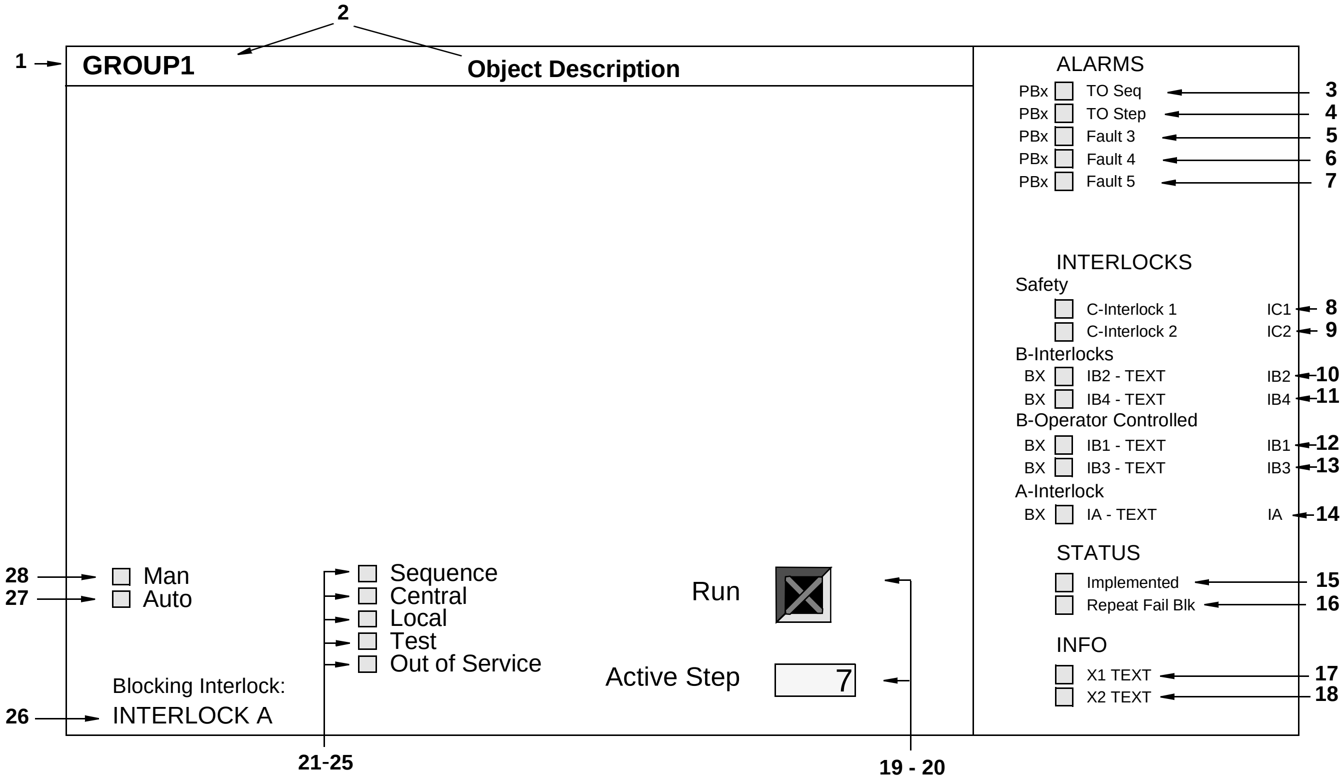
The display elements show the status and the controls of the process with different degrees of detail and is intended for the following displays:

- Object display
- Group display
- Object closeup display
- Process display

Examples of different display elements which could be used in these displays are given in the following section. A complete list of the prepared elements available is provided in the manual *AdvaCommand User Interface Reference Manual.*

Detailed information about alternative ways of presenting certain status and data within the displays such as color, flashing, blocked etc. is given in the reference manual.

Object display: GRPOB01

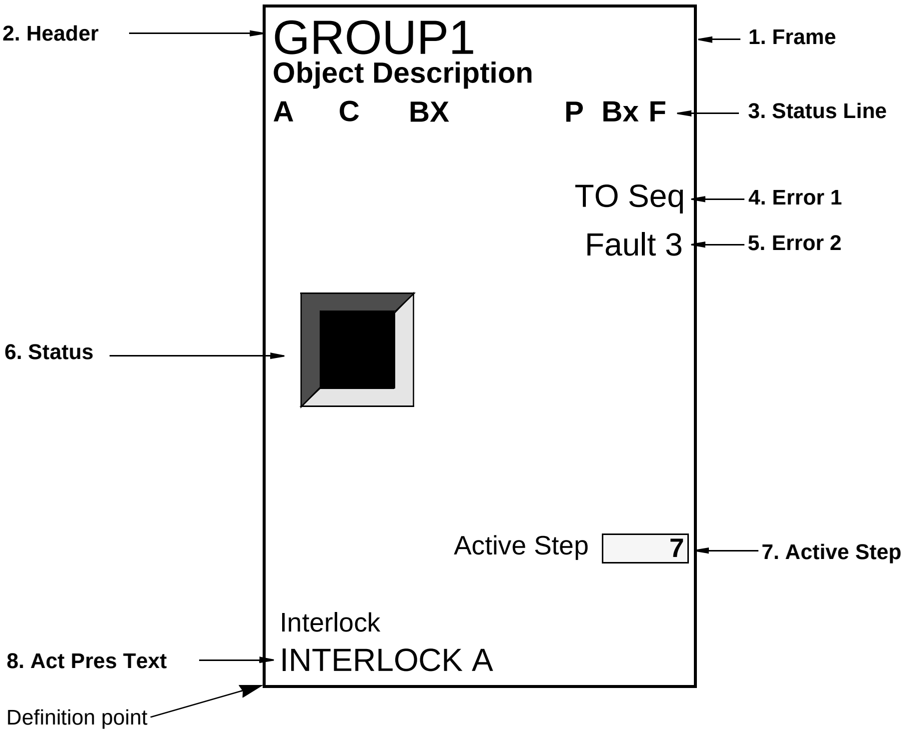


No	Description	Remarks
1	SelectFrame	
2	Header	Object name and description
3	ALARMS	
3.1	Printout Blocked	
3.2	Alarm Blocked	
3.3	Warning Indication	Unacknowledged alarm
3.4	Alarm 1 Text	TimeOut Sequence
4.1	Printout Blocked	
4.2	Alarm Blocked	
4.3	Warning Indication	Unacknowledged alarm
4.4	Alarm 2 Text	TimeOut Step
5.1	Printout Blocked	
5.2	Alarm Blocked	

No	Description	Remarks
5.3	Warning Indication	Unacknowledged alarm
5.4	Alarm 3 Text	Error text for fault 3
6.1	Printout Blocked	
6.2	Alarm Blocked	
6.3	Warning Indication	Unacknowledged alarm
6.4	Alarm 4 Text	Error text for fault 4
7.1	Printout Blocked	
7.2	Alarm Blocked	
7.3	Warning Indication	Unacknowledged alarm
7.4	Alarm 5 Text	Error text for fault 5
8	INTERLOCKS	
8.1	IC1 Active	Safety Interlock IC1 active
8.2	IC1 Active	Safety Interlock IC1

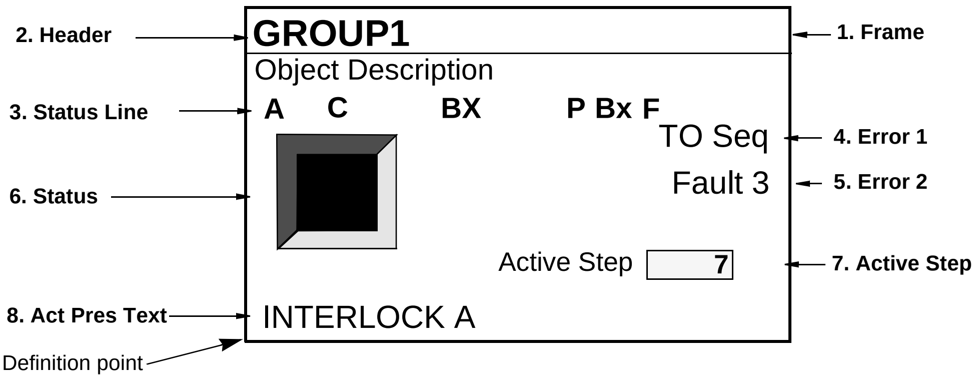
No	Description	Remarks
9.1	IC2 Active	Safety Interlock IC2 active
9.2	IC2 Text	Safety Interlock IC2
10.1	IB2 Blocked	Test mode. Interlock blocked
10.2	IB2 Active	Interlock IB2 active
10.3	IB2 Text	Interlock text IB2
11.1	IB4 Blocked	Test mode. Interlock blocked
11.2	IB4 Active	Interlock IB4 active
11.3	IB4 Text	Interlock IB4 text
12.1	IB1 Blocked	Interlock blocked
12.2	IB1 Active	Interlock IB1 active
12.3	IB1 Text	Interlock IB1 text
13.1	IB3 Blocked	Interlock blocked
13.2	IB3 Active	Interlock IB3 active
13.3	IB3 Text	Interlock IB3
14.1	IA Blocked	Test mode. Interlock blocked
14.2	IA Active	Interlock IA active
14.3	IA Text	Interlock IA text
15	STATUS	
15.1	Implemented Indication	
15.2	Implemented Text	
16.1	Repeat Fail Blocked Indication	The motor can be controlled
16.2	Repeat Fail Blocked Text	
17	INFO	
17.1	X1 Indication	Unack external alarm X1
17.2	X1 Text	
18.1	X2 Indication	Unack external alarm X2
18.2	X2Text	
19.1	Run	
19.2	Status	
20.1	Active Step	
20.2	Active Step value	Value of current step
21.1	Sequence Indication	Controlled from groupstart
21.2	Sequence Text	
22.1	Central Indication	Controlled from operator’s place
22.2	Central Text	
23.1	Local Indication	Controlled from local panel
23.2	Local Text	
24.1	Test Indication	Jog running from motor place
24.2	Test Text	
25.1	Out of Service Indication	No control possible
25.2	Out of Service Text	
26.1	Blocking Interlock Text	
26.2	Act Pres Text	
27.1	Auto Indication	Auto mode. Process conditions control start/stop of motor
27.2	Auto Text	
28.1	Man Indication	Manual mode. Starorder affects the motor directly
28.2	Man Text	

Display Element for Group display: GRPGR01



No	Description	Remarks
1	SelectFrame	
2	Header	Object name and description
3	Status Line	
3.1	Mode	
3.2	Point of Control	
3.3	Interlock Indication	
3.4	Printout Blocked	
3.5	Alarm Blocked	
3.6	Fault Indication	
4	Error 1	
5	Error 2	
6	Status Indication	
7.1	Active Step Text	
7.2	Active Step value	
8.1	Interlock Text	
8.2	Act Pres Text	

Display Element for Object closeup display: GRPWI01

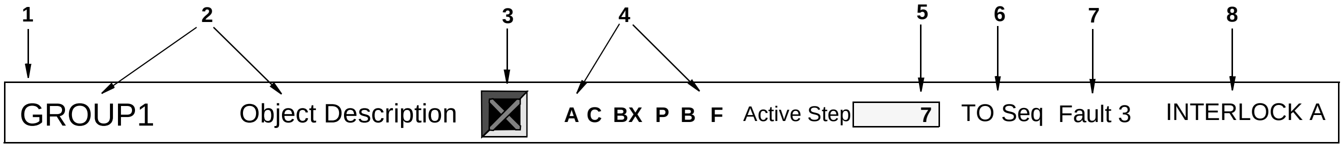


No	Description	Remarks
1	SelectFrame	
2	Header	Object name and description
3	Status Line	
3.1	Mode	
3.2	Point of Control	
3.3	Interlock Indication	
3.4	Printout Blocked	
3.5	Alarm Blocked	

No	Description	Remarks
3.6	Fault Indication	
4	Error 1	
5	Error 2	
6	Status	
7.1	Active Step Text	
7.2	Active Step value	
8	Act Pres Text	

Display Element for Process display:

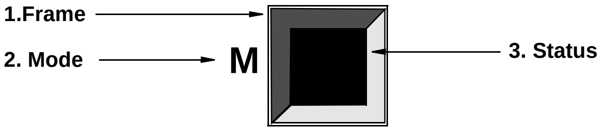
GRPSG01



No	Description	Remarks
1	SelectFrame	
2	Header	Object name and description
3	Status	
4.1	Mode	
4.2	Point Of Control	
4.3	Interlock Indication	
4.4	Printout Blocked	

No	Description	Remarks
4.5	Alarm Blocked	
4.6	Fault Indication	
5.1	Active Step Text	
5.2	Active Step value	
6	Error 1	
7	Error 2	
8	Act Pres Text	

GRPPD01



No	Description	Remarks
1	SelectFrame	
2	Mode or Control point	

No	Description	Remarks
3.1	Status Shape	
3.2	Status Color	

4.5.2 Dialog

The display screen is supplemented with a keyboard for operator communication with the functional unit/sequence.

The dialog is built up using both fixed and dynamic keys. The functions available with the dynamic keys are presented in a pattern of rectangles on the lower part of the display screen when the object is selected. The functional unit Sequence has a MIN dialog (Production level) and a MAX dialog (Maintenance level). The MIN dialog consists of an output dialog and one subdialog. The MAX dialog consists of an output dialog with 2 subdialog.

MIN dialog (No authority key)

Keyboard keys

Man	**Man**	Set to manual mode
Auto	**Auto**	Set to automatic mode
◇	**Start**	Start motor
▽	**Stop**	Stop motor
✓	**Acknowledge**	Acknowledge alarm
Object	**Object Display**	Select Object Display
📈	**Object Trend**	Select Object Trend
Dialog	**Previous Level**	Select Top Level dynamic keys

Dynamic keys, Top Level

D1 Sequence	D2 Central	D3 Local	D4	D5
D6 ⊟ Hold	D7 Instant Stop	D8	D9	D10 *Keyboard…*

Sequence	Set control to Sequence
Central	Set control to Central
Local	Set control to Local
Hold	Stop the stepping in the Sequence
Instant Stop	Stop the stepping in the Sequence
Keyboard…	Select Dynamic keys: Keyboard

Dynamic keys, Keyboard

D1 ■ Start	D2 Man	D3 Auto	D4	D5
D6 □ Stop	D7 Acknowledge	D8 Object Display	D9	D10

Start	Start motor
Man	Set to manual mode
Auto	Set to automatic mode
Stop	Stop motor
Acknowledge	Acknowledge alarm
Object Display	Select Object Display

MIN dialog (Authority key)

Keyboard keys

Man	**Man**	Set to manual mode
Auto	**Auto**	Set to automatic mode
◇	**Start**	Start motor
▽	**Stop**	Stop motor
✓	**Acknowledge**	Acknowledge alarm
Object	**Object Display**	Select Object Display
Dialog	**Previous Level**	Select Top Level dynamic keys
Dialog	**Next Block**	Page between the dynamic key sets (Keyboard and Block)

Dynamic keys, Top Level

D1 Sequence	D2 Central	D3 Local	D4 Test	D5 Our of Service
D6 ⊟ Hold	D7 Instant Stop	D8 *Block…*	D9	D10 *Keyboard…*

Sequence	Set control to Sequence
Central	Set control to Central
Local	Set control to Local
Test	Set control to Test
Out of Service	Set control to Out of Service
Hold	Stop the stepping in the Sequence
Instant Stop	Stop the stepping in the Sequence
Block…	Select Dynamic keys: Block
Keyboard…	Select Dynamic keys: Keyboard

Dynamic keys, Keyboard...

D1 ■ Start	D2 Man	D3 Auto	D4	D5
D6 □ Stop	D7 Acknowledge	D8 Object Display	D9	D10

Start	Start motor
Man	Set to manual mode
Auto	Set to automatic mode
Stop	Stop motor
Acknowledge	Acknowledge alarm
Object Display	Select Object Display

Dynamic keys, Block...

D1	D2 Block Alarm	D3 Block Printout	D4	D5 Override Interlock
D6	D7 Deblock Alarm	D8 Deblock Printout	D9	D10 Normal Interlock

Block Alarm	Block alarm indication
Block Printout	Block event printout
Override Interlock	Override interlocking
Deblock Alarm	Deblock alarm indication
Deblock Printout	Deblock event printout
Normal Interlock	Interlocking

4.5.3 Event Handling

The data base element MMCX has a number of properties which affect alarm and event handling for GROUP. These values can be changed during the circuit design.

The Event Treat defines the event handling in the operator station such as audible alarms, text combinations in event and alarm list, colors, different handling for 0 -->1 and 1 -->0 transitions.

How different events from GROUP is treated is given in Table 4-1 below. This also specifies which event handling is obtained as default and which alternative standard functions are available.The event handling can be blocked with I1_EVBLK and I2_EVBLK. Note that O_EVBLK has no function.

The Operator Station is provided with Event Treat data base elements to define the correct event handling.

Event description with standard texts

Table 4-1. Event Texts for GROUP

Events	Event Treat pointer	Property text		Event text	
		Standard text	Line No in Event Treat	Standard text	Line No in Event Treat
Timeout sequence	I1_TR	Timeout G	1	Alarm	1
Timeout sequence acknowledged	I1_TR	Timeout G	1	Normal	2
Timeout step	I1_TR	Timeout S	2	Alarm	1
Timeout step acknowledged	I1_TR	Timeout S	2	Normal	2
User defined Fault 3 coming	I1_TR	Fault 3	3	Alarm	1

Table 4-1. Event Texts for GROUP

Events	Event Treat pointer	Property text		Event text	
		Standard text	**Line No in Event Treat**	**Standard text**	**Line No in Event Treat**
User defined Fault 3 passing	I1_TR	Fault 3	3	Normal	2
User defined Fault 4 coming	I1_TR	Fault 4	4	Alarm	1
User defined Fault 4 passing	I1_TR	Fault 4	4	Normal	2
User defined Fault 5 coming	I1_TR	Fault 5	5	Alarm	1
User defined Fault 5 passing	I1_TR	Fault 5	5	Normal	2
Position A reached	I1_TR	Pos A	8	Alarm	1
Position A left	I1_TR	Pos A	8	Normal	2
Position B reached	I1_TR	Pos B	12	Alarm	1
Position B left	I1_TR	Pos B	12	Normal	2
Control mode Local On	I2_TR	Local	2	On	1
Control mode Local Off	I2_TR	Local	2	Off	2
Control mode Test On	I2_TR	Test	3	On	1
Control mode Test Off	I2_TR	Test	3	Off	2
Control mode Standby On	I2_TR	Standby	4	On	1
Control mode Standby Of	I2_TR	Standby	4	Off	2f
Control mode Sequence On	I2_TR	Sequence	5	On	1
Control mode Sequence Off	I2_TR	Sequence	5	Off	2
Op. mode Auto On	I2_TR	Auto	6	On	1
Op. mode Auto Off	I2_TR	Auto	6	Off	2
Interlock IC1 On	I2_TR	CInterl1	9	On	1
Interlock IC1 Off	I2_TR	CInterl1	9	Off	2
Interlock IC2 On	I2_TR	CInterl2	10	On	1
Interlock IC2 Off	I2_TR	CInterl2	10	Off	2
Interlock IB1 On	I2_TR	BInterl1	13	On	1
Interlock IB1 Off	I2_TR	BInterl1	13	Off	2
Interlock IB2 On	I2_TR	BInterl2	14	On	1
Interlock IB2 Off	I2_TR	BInterl2	14	Off	2
Interlock IB3 On	I2_TR	BInterl3	11	On	1
Interlock IB3 Off	I2_TR	BInterl3	11	Off	2

Table 4-1. Event Texts for GROUP

Events	Event Treat pointer	Property text		Event text	
		Standard text	**Line No in Event Treat**	**Standard text**	**Line No in Event Treat**
Interlock IB4 On	I2_TR	BInterl4	12	On	1
Interlock IB4 Off	I2_TR	BInterl4	12	Off	2
Interlock IA On	I2_TR	AInterl	15	On	1
Interlock IA Off	I2_TR	AInterl	15	Off	2

4.5.4 Group Alarm

It is possible to connect the functional unit GROUP to the group alarm function. This function makes it possible to gather alarms and events from related process signals and objects into Alarm Groups. These groups are then treated as objects.

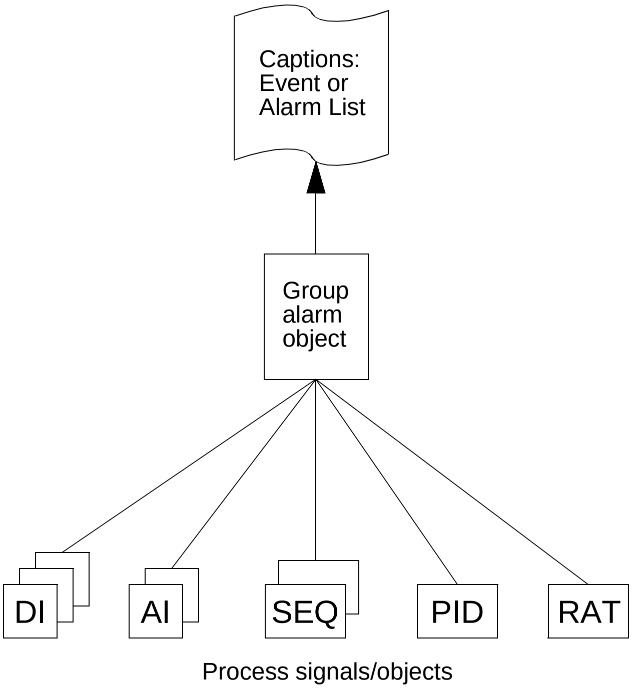


Figure 4-2. Group Alarm

The group alarm function consists of four parts as follows:

- Grouping of process alarms and events with a view to creating pilot alarms and events.
- Blocking of groups
- Automatic acknowledgement
- External alarm outputs

For a more detailed description of the group alarm function, reference is made to the manual *AdvaCommand Basic Functions User's Guide*.

4.6 Application Examples

The figure below illustrates, in principle, the use of the functional unit GROUP together with an appropriate PC circuit to start a number of motors via MOTCON. They are stopped at the same time by the operator's stop command.

The following process display for certain conveyor belts is suitable for the example.

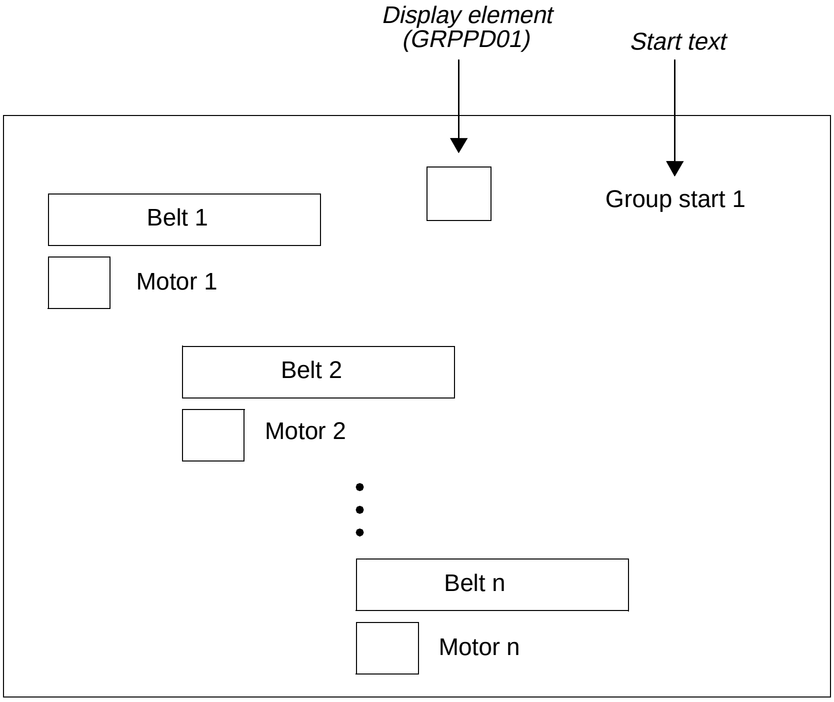


Figure 4-3. Illustration of a process display layout for a group of objects

Corresponding PC circuit

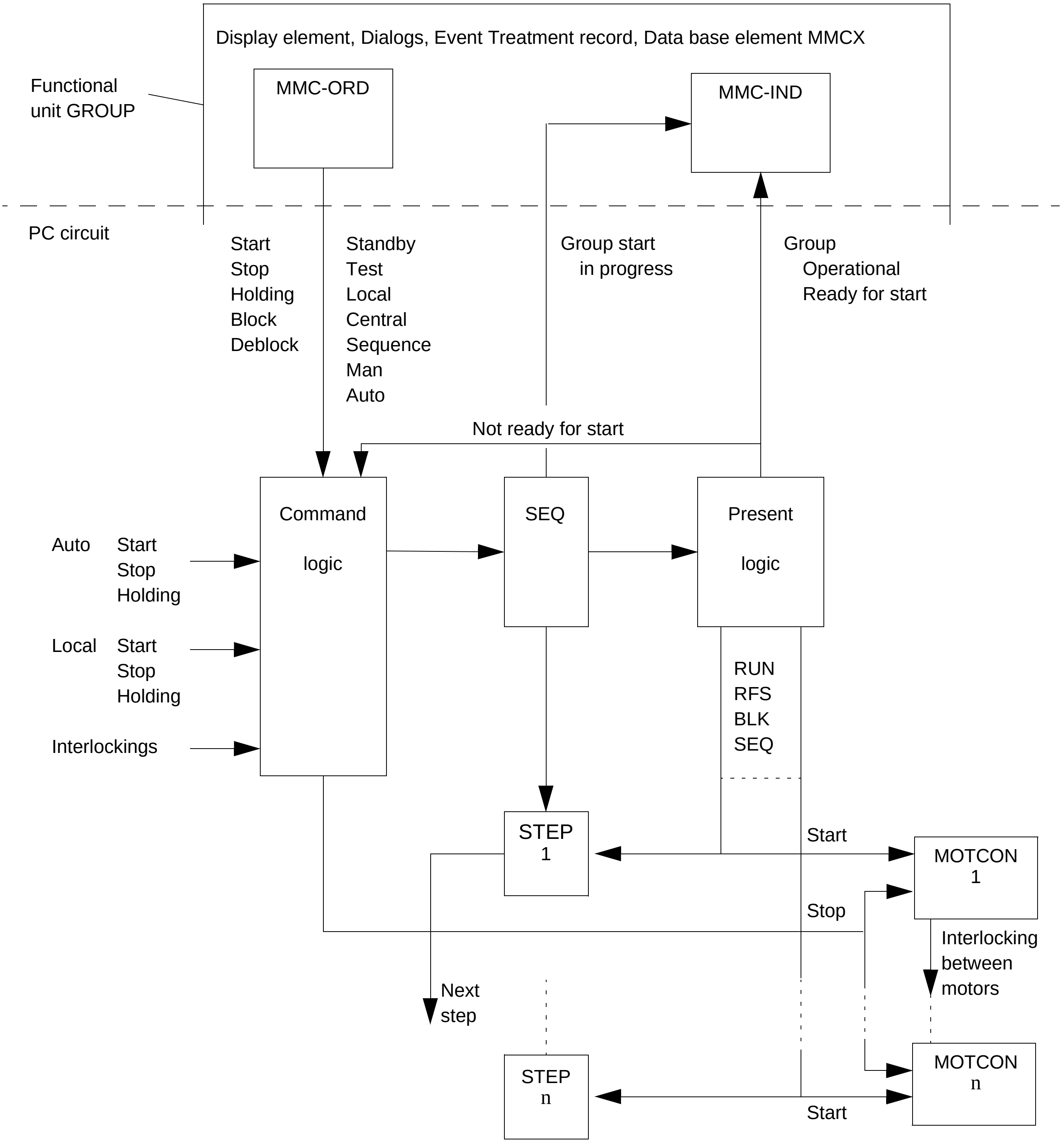


Figure 4-4. A principle block diagram of the application program

The group start circuit can be controlled both centrally from the display screen and keyboard and locally. It is presumed further that the circuit can be handled from transducers in the process in the control mode Auto.

The switching-over of points of control and control modes are intended to be performed centrally.

Interlockings which prevent starting or stopping of the group are suggested.

The start sequence is built up of the PC element SEQ and a STEP module for each motor in the group. The first step can, if required, control a warning siren instead of the function shown.

Each step initiates the next step.

The presentation logic evaluates the common status of the group. Indications that the group is in operation or that a start-up is in progress are given.

For the group to be Ready for Start all motors are to be in either the point of control Sequence and be Ready for Start or at Standby (not all however). Ready for Start= (Sequence and Ready for Start) or Standby.

I.e. the condition for start of the group is evaluated before any motor has been started. This simplifies condition generation for each step in the sequence at the same time that it prevents unnecessary starts which cannot be completed for the complete group.

For the group to be In operation, all motors are to be either in the status RUN or in the STANDBY point of control.

Interlockings are often necessary between the motors irrespective of whether they are to be group-started or not. For example, in the case of conveyor belts which succeed each other, a belt cannot be in operation if the succeeding belt is stationary. This means that Motor 1 in such a case is the last motor in the transport chain and is thus to start first. Note that these interlockings may not cause the group to be not Ready for Start as a group start function. MOTCON has a special interlocking input for this designated IA.

Presentation of a Group Start could be done in the following way:

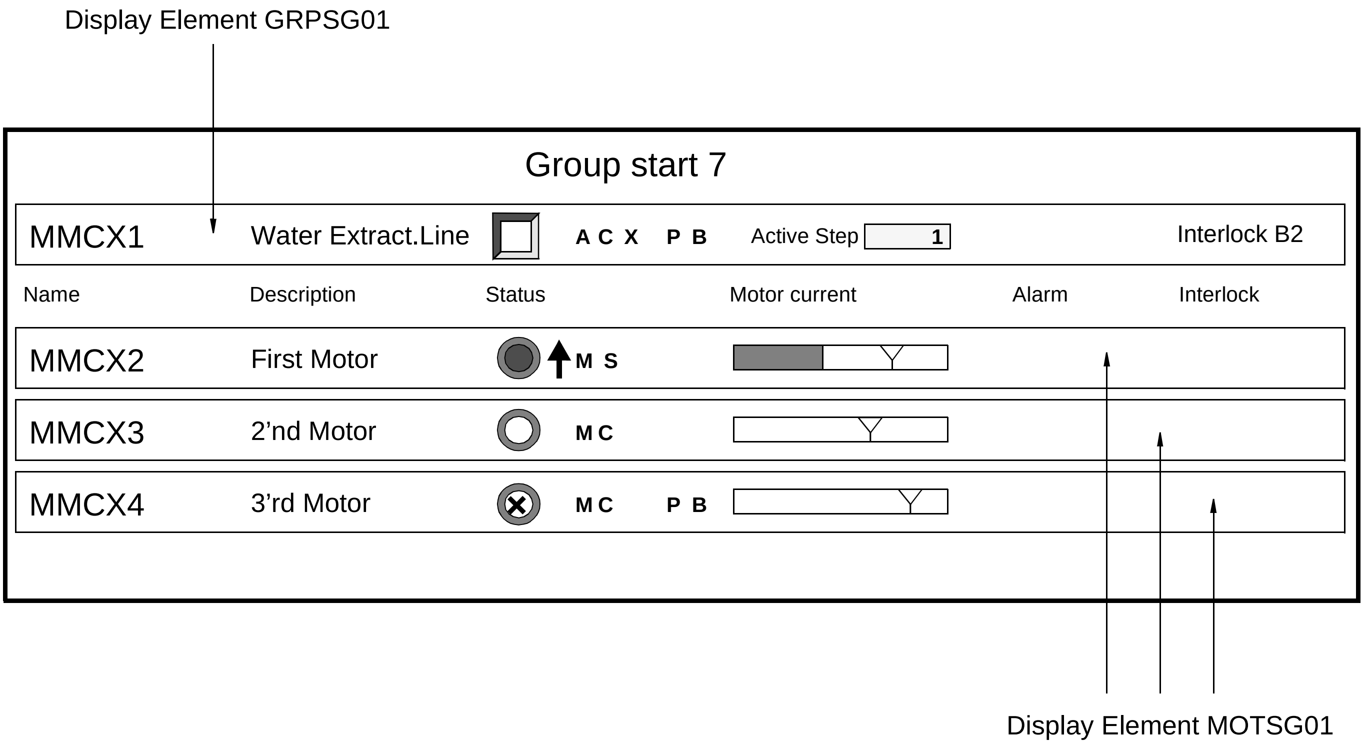


Figure 4-5. Group display layout on operators screen

3BSE 003 851R0001 Rev A

March 1996